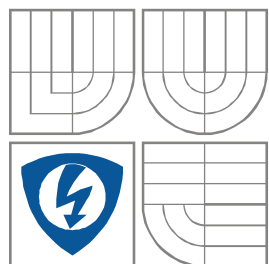


**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ**  
**ÚSTAV RADIOELEKTRONIKY**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF RADIO ELECTRONICS

# **POKROČILÁ SEGMENTACE OBRAZU PRO 3D ZOBRAZENÍ**

ADVANCED PICTURE SEGMENTATION FOR 3D VIEW

**DIPLOMOVÁ PRÁCE**  
MASTER'S THESIS

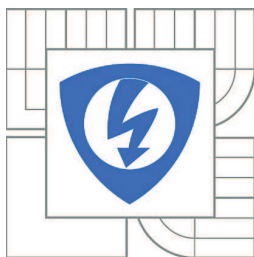
**AUTOR PRÁCE**  
AUTHOR

Bc. Tomáš Baletka

**VEDOUCÍ PRÁCE**  
SUPERVISOR

Ing. Libor Boleček

BRNO, 2012



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav radioelektroniky

# Diplomová práce

magisterský navazující studijní obor  
**Elektronika a sdělovací technika**

**Student:** Bc. Tomáš Baletka

**ID:** 106361

**Ročník:** 2

**Akademický rok:** 2011/2012

## NÁZEV TÉMATU:

**Pokročilá segmentace obrazu pro 3D zobrazení**

## POKYNY PRO VYPRACOVÁNÍ:

Seznamte se se základními a pokročilými metodami segmentace obrazu a rozpoznání objektu v obraze, založenými na různých přístupech. Vypracujte přehled perspektivně využitelných metod, diskutujte možnosti jejich kombinace.

Na základě vypracovaného přehledu vyberte vhodné metody a vytvořte algoritmus segmentace. Vytvořte uživatelské prostředí aplikace a implementujte segmentační postup.

Vyvinutou aplikaci důkladně otestujte na vhodně vybraných snímcích. Výsledky využijte pro tvorbu anaglyfického 3D zobrazení.

## DOPORUČENÁ LITERATURA:

[1] JAN, J. Medical Image Processing, Reconstruction and Restoration - Concepts and Methods. Boca Raton, FL, USA: CRC Press, Taylor and Francis Group, 2005. 760 s. ISBN: 0-8247-5849- 8.

**Termín zadání:** 6.2.2012

**Termín odevzdání:** 18.5.2012

**Vedoucí práce:** Ing. Libor Boleček

**Konzultanti diplomové práce:**

**prof. Dr. Ing. Zbyněk Raida**  
*Předseda oborové rady*

## UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **ABSTRAKT**

Diplomová práce pokročilé segmentace obrazu pro 3D se zabývá segmentací obrazu a anaglyfického 3D zobrazení. V teoretické části diplomové práce byly popsány různé přístupy používané k segmentaci obrazu a úzce související metody zpracování obrazu. V navazující praktické části byla provedena implementace vybraných metod a vytvořena uživatelsky příjemná aplikace. Hlavním cílem programu je identifikace výrazných objektů v obraze. Za účelem segmentace byly implementovány metody založené na metodě k-means, na metody kontur a na růstu ze semínka.

Program je vytvořen v prostředí Visual Studio 2008 a napsán v programovacím jazyku C++. Vstupem i výstupem programu je obraz v různých formátech (JPG, BMP, TIFF).

## **KLÍČOVÁ SLOVA**

Segmentace obrazu, Barevný model, Histogram, K-means, Kontury, Růst ze semínka, Anaglyf

## **ABSTRACT**

The thesis advanced image segmentation for 3D image deals with segmentation and anaglyph 3D views. In the theoretical part of the thesis describes the different approaches were used to image segmentation and closely related methods of image processing. In the following practical part was the implementation of selected methods and created user-friendly applications. The main objective of the program is to identify significant objects in the image. For the purpose of segmentation methods have been implemented based on k-means method, the method of contour and the growth of seeds.

The program is created in Visual Studio 2008 and written in C + +. The input and output is the image in various formats (JPG, BMP, TIFF).

## **KEYWORDS**

Image segmentation, Color model, Histogram, K-means, Countours, Growth of seeds, Anaglyph

BALETKA, T. *Pokročilá segmentace obrazu pro 3D zobrazení*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2012. 47 s. Vedoucí diplomové práce Ing. Libor Boleček.

## PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma Pokročilá segmentace obrazu pro 3D zobrazení jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne .....

.....

(podpis autora)

## PODĚKOVÁNÍ

Děkuji vedoucímu semestrálního projektu Ing. Liboru Bolečkovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne .....

.....

(podpis autora)

# OBSAH

<b>Úvod</b>	<b>1</b>
<b>1 Metody Předzpracování obrazu</b>	<b>2</b>
1.1 Jasové transformace .....	2
1.1.1 Lokální transformace .....	2
1.1.2 Bodové jasové transformace .....	2
1.2 Geometrické transformace .....	3
1.3 Matematická morfologie .....	3
<b>2 Barevné modely</b>	<b>4</b>
2.1 RGB .....	4
2.2 HSV, HLS .....	5
2.2.1 HSV .....	5
2.2.2 HLS .....	6
2.3 YCrCb .....	7
2.4 XYZ .....	7
2.5 Luv, Lab .....	7
<b>3 Přehled segmentací</b>	<b>9</b>
3.1 Parametrická segmentace .....	9
3.1.1 Segmentace podle jasu .....	9
3.2 Hranově založená segmentace .....	10
3.2.1 Hranové detektory .....	10
3.2.2 Detekce první derivací .....	10
3.2.3 Detekce druhou derivací .....	11
3.2.4 Houghova transformace .....	12
3.3 Regionově orientovaná segmentace .....	14
3.3.1 Segmentace růstem oblastí .....	14
3.3.2 Segmentace dělením a slučováním oblastí .....	14
3.3.3 Segmentace Watershed .....	15
3.3.4 Neuronové sítě .....	16
3.4 Segmentace metodou pružných a aktivních kontur .....	17

3.4.1	Parametricky pružné kontury .....	17
3.4.2	Metoda geometrických kontur .....	18
3.4.3	Metoda aktivních kontur .....	19
<b>4</b>	<b>Základní struktura programu Seganag</b>	<b>20</b>
4.1	Knihovna OpenCV .....	20
4.2	Vývojový diagram .....	20
4.3	Grafické rozhraní .....	22
4.4	Popis části aplikace .....	22
4.4.1	Načtení, zobrazení a převedení snímku na barevný model .....	22
4.4.2	Zobrazení histogramu složek .....	23
4.4.3	Funkce určující vhodný model pro segmentaci .....	23
4.4.4	Segmentační metoda využívající metodu Kmeans .....	25
4.4.5	Segmentační metoda růstu ze semínka .....	27
4.4.6	Segmentační metoda kontur .....	28
4.4.7	Anaglyf .....	31
4.4.8	Uložení obrázku .....	31
<b>5</b>	<b>Uživatelská příručka</b>	<b>32</b>
5.1	Hlavní obrazovka .....	32
<b>6</b>	<b>dosažené výsledky</b>	<b>36</b>
6.1	Segmentace založená na metodě Kmeans .....	37
6.2	Segmentace založená na metodě semínko .....	40
6.3	Segmentace založená na metodě kontur .....	41
6.4	Anaglyf .....	42
	<b>Závěr</b>	<b>43</b>
	<b>Literatura</b>	<b>45</b>
	<b>Seznam symbolů, veličin a zkratk</b>	<b>47</b>

# ÚVOD

V dnešním světě počítačové techniky hraje důležitou roli segmentace obrazů. Jako takovou ji můžeme označit za samostatnou linii zpracování obrazu. Která čerpá doposud nabyté poznatky ze zpracování obrazu. Samotná segmentace obrazu je dnes důležitou úlohou automatického zpracování obrazu. Je úzce spojena s úkolem rozpoznání objektů v obraze. V současnosti je nedílnou součástí počítačového zpracování obrazu. Využívá se v mnoha odvětvích, jako např. medicína, počítačová grafika, digitální fotografie, bezpečnostní systémy a v různých oblastech průmyslu.

Úkol segmentace je rozdělovat jednotlivé části obrazu na objekty. Výsledky jsou pak používány například pro rozpoznávání nebo identifikaci objektů. Cílem segmentace je rozdělení obrazu na jednotlivé objekty, které se v ideálním případě shodují s objekty reálného světa. Toho můžeme dosáhnout aplikováním některé segmentační metody, které byly v průběhu let publikovány. Přestože je již navrženo několik desítek metod, pořád je toto téma aktuální a vyvíjejí se další segmentační postupy. Bohužel segmentace je náročný úkol a neexistuje univerzální metoda, která by fungovala pro všechny obrazy. Proto při výběru vyhovující segmentační metody je účel, pro který budeme metodu využívat. Například chceme-li analyzovat snímky v medicíně, využijeme jiná metody než při analyzování objektů (osob, budov) v obraze.

Cílem teoretické části diplomové práce je vytvořit přehled základních a pokročilých metod segmentace, založené na různých přístupech. V navazující praktické části práce jsou implementovány některé segmentační metody. Je vytvořeno grafické uživatelské prostředí. Výstupem programu je i návrh a zobrazení 3D snímku anaglyfickou metodou. Diplomová práce je rozdělena do 6 kapitol. V první kapitole jsou zmíněny metody předzpracování, které mají za účel odstranit šum, anebo potlačení nerovnoměrného osvětlení obrazu. V další kapitole jsou popsány barevné modely obrazu. Volba vhodného modelu má nemalý vliv na výsledek segmentace, protože každý model popisuje snímek jiným způsobem. Například u modelu RGB je výsledný obrazový bod tvořen poměrem červené, zelené a modré barvy. U HSV modelu je to poměr barevného tónu, sytosti barvy a hodnotou jasu. V následující kapitole je zmíněn přehled základních i pokročilých segmentačních metod. Dozvíme se něco o parametrické segmentaci, hranově založené segmentaci, regionově orientované segmentaci a segmentaci pružných a aktivních kontur. Čtvrtá kapitola je věnovaná základní struktuře programu SegAnag, kde jsou návrhy jednotlivých postupů programu. Další kapitola obsahuje jednoduchou uživatelskou příručku, kde se popisuje ovládání programu. Poslední kapitola je věnována dosaženým výsledkům. Otestování vlivu změny různých parametrů na výslednou segmentaci. Vstupními daty vytvořené aplikace je obrázek v různých formátech (JPG, BMP, TIFF). Výstupem je buď barevný obraz rozdělený do jednotlivých segmentů, nebo anaglyfický snímek. Vytvořená aplikace poskytuje uživateli zvolit ze tří metod segmentace, u kterých lze měnit jejich vstupní parametry a tím měnit výsledný vysegmentovaný obraz. Aplikace je naprogramována v jazyku C++ při použití knihovny openCV v programu Visual Studio 2008.



# 1 METODY PŘEDZPRACOVÁNÍ OBRAZU

Samotné segmentaci obrazu často předchází předzpracování obrazu, které může pomoci k dosažení lepších výsledků segmentace. Cílem předzpracování je hlavně potlačit šum, odstranit zkreslení a zvýraznit důležité hrany obrazu. Vstupem i výstupem předzpracování je obraz, který reprezentují různé hodnoty (např. jasu) pixelů. Tyto hodnoty bývají pro sousední pixely podobné, proto se redukují o nadbytečné údaje. Různé metody předzpracování vedou na transformaci vstupního obrazu. Metody předzpracování nejsou univerzální a použití stejné metody na rozdílné obrazy může přinést kvalitativně velmi odlišné výsledky. Z toho důvodu bude mít uživatel ve vytvářené aplikaci na výběr s několika metod předzpracování a bude mít sám možnost volby.

## 1.1 Jasové transformace

Jedna ze základních metod je jasová transformace. Může být buď bodová, kde se upravuje jen samotná hodnota jasu, nebo lokální, která bere v úvahu okolí zkoumaného bodu. Jedné z lokálních transformací se říká filtrace, která se může provádět buď ve frekvenční, nebo časové oblasti. Převody mezi časovou a frekvenční oblastí se používá nejčastěji kosínová, vlkoová a Fourierova transformace.

### 1.1.1 Lokální transformace

Využívají pro výpočet jasu bodu ve výstupním obrazu jen lokální okolí odpovídajícího bodu ve vstupním obrazu. Toho se docílí podle požadovaného výstupního obrazu pomocí dvou odlišných technik. Za prvé vyhlazení obrazu, tady se potlačují vyšší frekvence. Tím se zajistí potlačení náhodného šumu i jiných náhlých změn. Za druhé detekci hran, tahle metoda je přesně inverzní k první metodě. Zdůrazňuje vyšší frekvence, tím pádem zvýrazní obrazové elementy, kde se jasové funkce náhle mění, mají velký modul gradientu. Bohužel se zvýrazňuje i šum. Vychází z principu, že hrana je místo náhlé změny jasu. Tyto místa se musí zdůraznit.

### 1.1.2 Bodové jasové transformace

Tato metoda předzpracování se zabývá jasovou korekcí nebo transformací vstupního obrazu. Pro úpravu pixelů obrazu používáme jenom změnu hodnoty jasového pixelu. Jeden typ této transformace se nazývá jasová korekce, která využívá nedokonalosti snímacích zařízení. Tato nedokonalost se projevuje různou jasovou citlivostí na snímání objekt. Vlivem různého osvětlení předmětu vznikají systematické chyby.

Jako druhý typ se používá transformace jasové stupnice. To znamená, že jen určitá hodnota jasu je transformována na jinou hodnotu. Tento typ transformace nesouvisí na pozici jasových hodnot. Vstupní stupnice jasu obrazu a je transformována na stupnici:  $v=T(u)$ . Příklady těchto transformací můžou být inverze, prahování, ekvalizace histogramu (algoritmus, který změnil rozložení intenzity v obraze na rovnoměrné a o zvýšení kontrastu) atd.

## 1.2 Geometrické transformace

Tato metoda předzpracování má za cíl odstranění geometrických zkreslení v obraze. Docílí se toho změnou rozlišení, posunutí nebo zkosení 2D obrazu. K výsledku geometrické transformace se používají 2 kroky, a to plošná transformace (transformace souřadnic bodů) a jasová transformace (aproximace jasové funkce). Při transformaci mohou nastat problémy, jednak že se nepřihadí celočíselným souřadnicím ve vstupním obraze celočíselné souřadnice ve výstupním obraze. Dále může nastat, že část původního obrazu bude ležet mimo nový obraz a většina provedených transformací není invertovatelná na původní obraz. [3]

## 1.3 Matematická morfologie

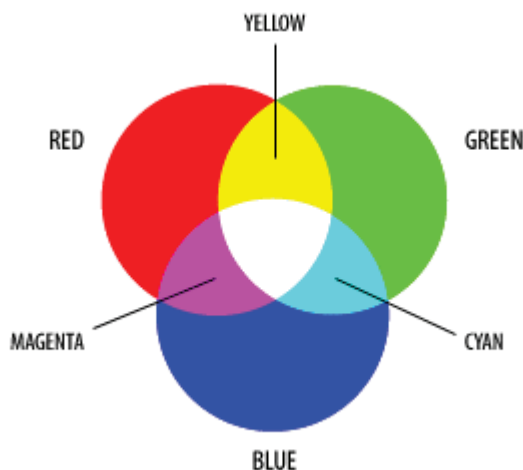
Další možnou technikou, která se používá je matematická morfologie. Využívá se pro předzpracování, a to hlavně v odstranění šumu a zjednodušení tvaru objektů, zdůraznění struktury objektů (např. ztenčování, zesilování, označování objektů) a popis objektů číselnými charakteristikami (plocha, obvod). Základem této metody jsou tvary objektů, které se při transformaci nemění. Jako realizace se používá relace obrázků s jinou menší bodovou množinou, nazývaní se strukturní element, kterým systematicky pohybujeme v obraze. Mezi základní operace při této metodě se používají dilatace, eroze, otevření, uzavření. [3]

## 2 BAREVNÉ MODELÝ

V této kapitole si představíme některé barevné modely, které se používají pro reprezentaci obrázku při segmentaci. Vhodně zvoleným barevným modelem můžeme ovlivnit výsledek segmentace. Každý barevný model je charakterizován pomocí určitých parametrů. Jako na příklad jasem, svítivosti, sytostí, světlostí. Barevné modely používají základní barvy a jejich skládání do výsledné barvy. Barevné modely můžeme dělit podle míchání barev na aditivní, kde se pracuje se světelnými zdroji barev, a subtraktivní, kde barevný model pracuje s odrazem světla.

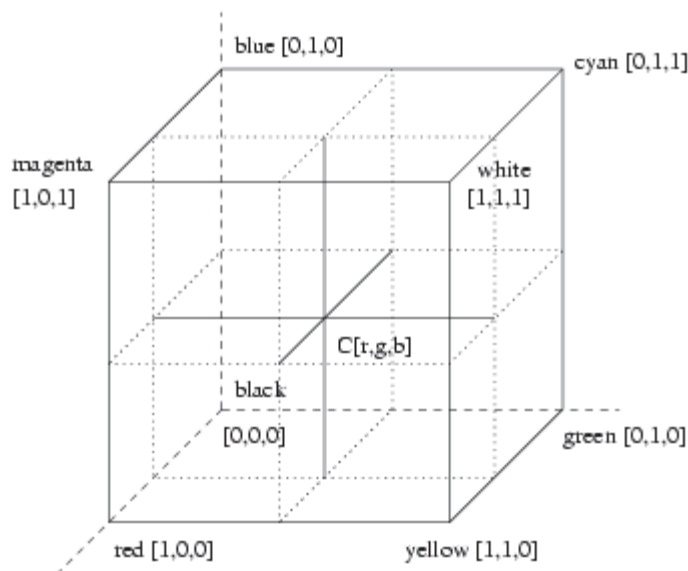
### 2.1 RGB

Tento model patří do aditivního míchání barev, tzn. jestliže jsou jednotlivé složky barevného modelu větší, tím bude výsledná barva světlejší. Toho se využívá například v zobrazení barev v monitoru nebo televizi.



Obr 2.1: Aditivní skládání barev u RGB

Model RGB je vyjádřen jako vážený součet tří základních barev. Červené (Red), zelené (Green) a modré (Blue). Jako vyjádření podílu velikosti dané barvy se nejčastěji používá interval  $<0,1>$ . V počítačové grafice je tento interval intenzity základní barvy převeden na celočíselný interval  $<0,255>$ . Libovolný barevný bod je potom reprezentován 24 bity, jelikož máme 3 složky po 8 bitech. Takto můžeme zobrazit až  $256^3 = 16\,777\,216$  barevných odstínů. Barvy zobrazené 24 bity nazýváme jako pravé barvy (true colors). RGB model si můžeme představit jako jednotkovou krychli (obr. 2), kdy v počátku  $(0,0,0)$  máme černou barvu a v protilehlém vrcholu  $(1,1,1)$  bílou barvu.



Obr. 2.1: RGB model krychle[8]

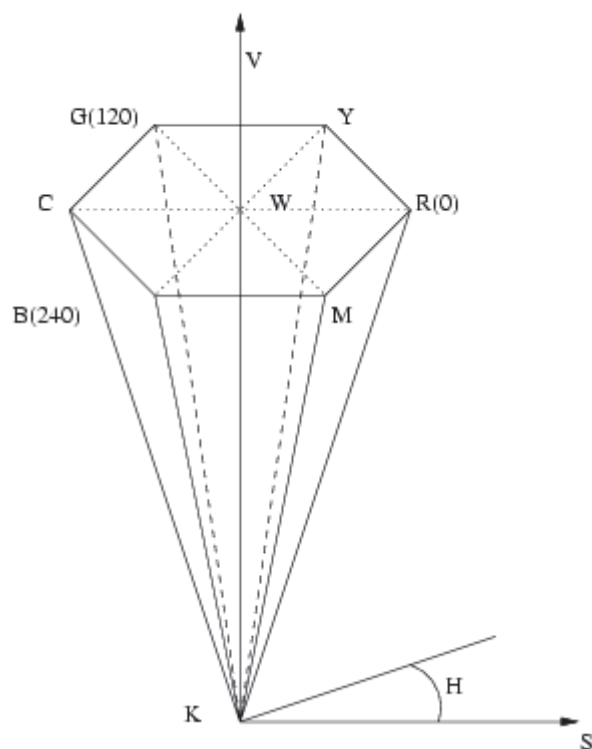
## 2.2 HSV, HLS

Protože mnoho lidí si nedokáže představit, jaká barva vznikne smícháním několika jiných barev zavádíme barevné modely, které tento problém řeší. Používají pro vyjádření barvy jinou techniku. [8] Tato technika je založena na intuitivním způsobu míchání barev.

### 2.2.1 HSV

Výsledná barva v HSV modelu je definovaná pomocí barevného tónu (Hue), sytosti (Saturation) a jasové hodnoty (Value). Jako geometrickou reprezentaci si můžeme představit pravidelný šestiboký jehlan (obr 3). Kde barevný tón představuje úhel převládající barvy a může nabývat  $0-360^\circ$ . Jasová hodnota pak udává výšku jehlanu a dané množství bezbarvého světla. Sytost vyjadřuje relativní vzdálenost bodu od osy jehlanu a znamená čistotu barvy, nebo také příměs jiných barev. Sytost i jasová hodnota nabývají hodnot 0 až 1.

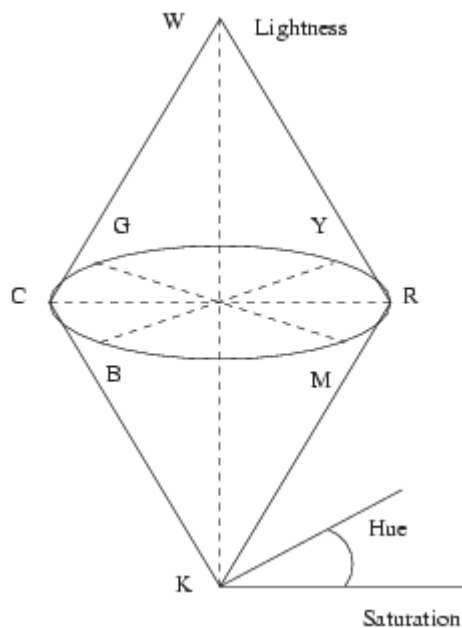
Někdy je nevýhodný pro svůj jehlanovitý tvar, který způsobuje, že ve vodorovném řezu se musí bod o konstantní hodnotě S pohybovat při změně H po dráze ve tvaru šestiúhelníku a nikoliv po kružnici, jak by bylo přirozené. Dalším záporným jevem je nesymetrie modelu z hlediska jasu. [9]



Obr. 2.2: Geometrická reprezentace modelu HSV [8]

### 2.2.2 HLS

V tomto barevném modelu je barva reprezentovaná barevným tónem (Hue), světlostí (lightness) a sytostí (Saturation). Geometrické vyjádření modelu jsou podstavami spojené dva kužely (obr. 4). Barevný tón představuje jako v modelu HSV převládající barvu. Světlost udává výšku a velikost achromatické složky. Místo, kde jsou spojené kužely odpovídá 0,5 světlosti. Saturace jako u HSV je relativní vzdálenost bodu od osy, kde když je bod na plášti jehlanu má hodnotu 1. HLS model řeší některé nedostatky HSV modelu.



Obr. 2.3: Geometrická reprezentace modelu HLS[8]

## 2.3 YCrCb

Model je založen na subtraktivním mísení barev. Barvy jsou popsány tříprvkovým vektorem  $[Y, C_b, C_r]$ , kde  $Y$  je jasová složka a  $C_b$ ,  $C_r$  jsou barevné složky. Barevné složky se používají v rozsahu od -0.5 do +0.5.

Tento model se využívá v oblasti digitálních fotoaparátů a speciálních videozařízení (pomaloběžná televize). Využívá se zejména možnosti jednoduše oddělit černobílý obraz, který je reprezentovaný jasovou složkou a informací o barvě, která je reprezentována dvěma chroma parametry (sytnost červené a modré barvy). [13]

## 2.4 XYZ

XYZ je jeden z prvních matematicky definovaných barevných prostorů. Byl definován ve studii o vnímání barev vytvořené mezinárodním úřadem ommission internationale de l'éclairage (zkráceně CIE) v roce 1931. Vychází z předchozí specifikace zvané RGB, která vzešla ze série experimentů.

Koncept vnímání barev ovšem může být rozdělen na 2 části – jas a barvu. Prostor XYZ byl záměrně navržen tak, že parametr  $Y$  vyjadřuje jas a vlastní barva je specifikována dvěma odvozenými parametry  $x$  a  $y$ . Tyto odvozené parametry lze spočítat ze všech tří trichromatických složek  $X$ ,  $Y$  a  $Z$ . Barevný prostor udaný parametry  $x$  a  $y$  se nazývá  $xyY$  nebo jen  $xy$  (bez parametru jasu). [14]

## 2.5 Luv, Lab

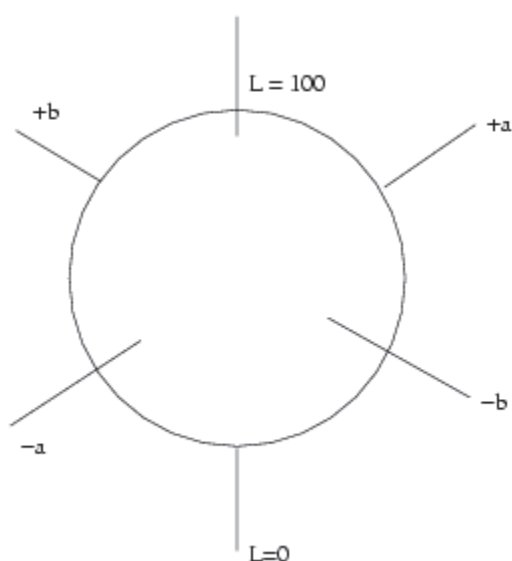
V roce 1976 byl model  $xyY$  upraven a vznikl model Luv (Uniform Color Space).

Vzdálenost dvou různých barev v tomto modelu lépe odpovídá lidskému vnímání rozdílu těchto barev.

Pro výpočet této vzdálenosti je dokonce definována metrika. Mějme dvě barvy  $[L_1, u_1, v_1]$  a  $[L_2, u_2, v_2]$  a bílý bod  $[L_0, u_0, v_0]$ . Potom rozdíl barev je definován jako: [8]

$$\Delta E = \sqrt{(L_1 - L_2)^2 + (\Delta u_1 - \Delta u_2)^2 + (\Delta v_1 - \Delta v_2)^2} \quad (2.1)$$

Model Luv je určen pro vyjadřování barev zdrojů z vlastním zářením, naproti tomu model CIE  $L^*a^*b^*$  (někdy nazýván Lab) je určen pro barvy materiálu. [8]



Obr. 2.4: Geometrická reprezentace Lab[8]

## 3 PŘEHLED SEGMENTACÍ

Segmentace obrazu je souhrnný proces, který se skládá z několika metod. Vstupní obraz je nutno obvykle nějakým způsobem upravit. Týká se to především odstranění šumu, anebo velkého počtu zanedbatelných detailů. Možnými metodami předzpracování se zabývá kapitola 1.

Dále už po předzpracování následuje samotný proces segmentace. Po segmentaci je nutno však provést většinou ještě dodatečné procesy, protože samotná segmentace nevytvoří zcela použitelné výsledky. Ty mohou být vytvořeny malými segmenty, které je pak potřeba spojit, nebo opomenou rozdělit objekty, které spolu nesouvisí.

Postupně si představíme některé metody na segmentaci obrazu. Segmentace se dělí na parametrickou, hranově a regionově orientovanou segmentaci a metody pružných a aktivních kontur.

### 3.1 Parametrická segmentace

Velká skupina segmentačních metod provádí segmentaci obrazu na základě vývoje zvoleného parametru. Parametry mohou být jas a jiné obrazové reprezentace bodu v obraze. Tento parametr se pak porovnává se segmentovaným obrazem. Pokud je tento parametr konstantní s určitou tolerancí, tak se konstantní oblast obrazu označí jako jeden segment, tzn. jeden objekt.

#### 3.1.1 Segmentace podle jasu

Velmi jednoduchá metoda, ve které zadáváme meze hodnoty jasu v obraze. Poté vstupní obraz  $f(i,j)$  prahujeme, tj. porovnáváme úroveň jasu jednotlivých pixelů obrazu s hodnotou zadaného prahu, na výstupní segmentovaný obraz  $g(i,j)$ . Výsledný obraz je binární, obsahuje hodnoty pixelu 0 nebo 1, dané tím jestli se nachází pod zadanou úrovní prahu nebo nad ní. [1]

$$g(i,j) = \begin{cases} 1 & \text{pro } f(i,j) > T \\ 0 & \text{jinak} \end{cases} \quad (3.1)$$

U dvojitého prahování místo jednoho prahu jsou dva. Výsledné vysegmentované oblasti se nachází mezi těmito hodnotami prahů.

$$g(i,j) = \begin{cases} 1 & \text{pro } f(i,j) \in \langle T_1, T_2 \rangle \\ 0 & \text{jinak} \end{cases} \quad (3.2)$$

Další modifikace vychází s obou dvou předešlých metod a nazývá se poloprahování, kdy se potlačí nezajímavé části vstupního obrazu a zbylé části výstupního obrazu zůstávají stejné jako odpovídající části vstupního obrazu.

$$g(i,j) = \begin{cases} f(i,j) & \text{pro } f(i,j) > T \\ 0 & \text{jinak} \end{cases} \quad (3.3)$$



Dalším způsobem je adaptivní prahování, které se liší od základního tím, že se liší hodnota prahu pro různé oblasti obrazu, představuje tak funkci lokálních parametrů obrazu. Obraz nejdříve rozdělujeme do několika částí, pak pro každou část najdeme prahy (mohou být interpolovány) a nakonec provedeme prahování pro každou část s jejím konkrétním prahem [1].

$$g(i,j) = \begin{cases} 1 & \text{pro } f(i,j) > D \\ 0 & \text{jinak} \end{cases}, \quad (3.4)$$

kde  $D$  je jistá množina úrovně jasu.

## 3.2 Hranově založená segmentace

Tato metoda vychází z detekce hran v obraze. Jedná se o jednu z důležitých částí nižšího zpracování obrazu. Hraný jsou definovány jako místa obrazu s prudkou změnou sledované veličiny. Většinou se sleduje jasová složka obrazu a její změny, protože lidské oko je citlivé na změny jasu. Hranu si můžeme představit jako vektor zadaný svou velikostí a směrem, který určuje vlastnost jednotlivých obrazových bodů. Při analýze obrazu pozorujeme skokové změny jasové složky, které představují hrany přechodu od jednoho objektu ke druhému.

V obraze můžeme nalézt různé typy hran. Průběhu ideální hrany odpovídá skok (step). V reálných obrazech však dochází ke změně jasu postupně, proto zde nacházíme šikmou hranu (ramp). Pokud se šikmý a skokový průběh hran objeví v obraze vedle sebe, vznikají ještě další dva typy hran: linie (line) a střecha (roof) [1].



Obr. 3.1: Typy hran [4]

Při detekci hran se používá algoritmus, který má tři části. První z nich je filtrování (odstranění šumu v obraze) následuje diferenciací (zdůraznění změny jasu v obraze) a nakonec detekci (lokalizuje nejvýznamnější body změny jasu).

### 3.2.1 Hranové detektory

Rozlišujeme dvě hlavní skupiny metod detekce hran. Tyto metody využívají první derivaci nebo druhou derivaci. Při použití první derivace je výsledek porovnáván s prahem, který určuje, jedná-li se o hrany, či nikoli. U druhé derivace je zaznamenán výskyt hrany jen v tom případě, je-li prostorová změna v polaritě druhé derivace dostatečně významná [4].

### 3.2.2 Detekce první derivací

Tato detekce používá rozdíly sousedních pixelů. Nejjednodušší je vypočítat derivaci samostatně pro sloupce a řádky. Sousední pixely, ze kterých se vypočítává gradient, vybíráme nejbližší pixely v horizontálním a vertikálním směru.

Výsledný gradient vypočteme pomocí [4]:

$$G(i, j) = \sqrt{G_R(i, j)^2 + G_S(i, j)^2}, \quad (3.5)$$

Kde  $G(i, j)$  je výsledný gradientní obraz,  $G_R(i, j)$  je gradient po detekci hran pro horizontální směr, a  $G_S(i, j)$  je gradient pro vertikální směr.

Gradient obrazu můžeme určit jako konvoluci hranových detektorů s obrazem. Hranové detektory jsou vlastně masky (filtry) určující jaké body a s jakou vahou budou použity pro výpočet výsledného gradientu obrazu. Pro lepší citlivost na šum je lepší použít filtry s větší velikostí okolí, např.  $5 \times 5$ ,  $7 \times 7$ .

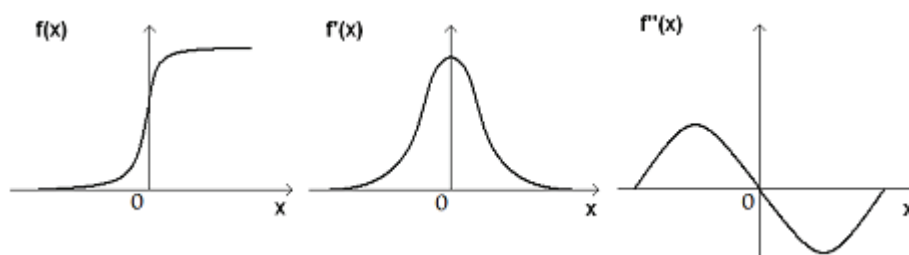
Většina operátorů pro zvýraznění hran byly vyvinuty heuristicky. Canny zvolil analytický přístup a jeho operátor je nastaven tak, aby co nejlépe vyhovoval následujícím podmínkám:

- dobrá detekce – poměr signál - šum by měl být co největší => nejmenší pravděpodobnost selhání výběru hrany,
- dobrá lokalizace – vybrané hranové body leží co nejbližně středu hrany,
- jedinečná odezva – existence jedinečné odezvy na skutečnou hranu,

Realizace Cannyho operátoru kvůli složitosti není možná, proto se používají různé způsoby pro jeho vytvoření, které se snaží co nejvíce přiblížit stanoveným podmínkám. Příkladem může být využití první derivace Gaussovske funkce, adaptivního prahování a techniky potlačení nemaximálních hodnot [4].

### 3.2.3 Detekce druhou derivací

Pomocí této metody nezkoumáme velikost gradientu, ale průchod nulou. Nalezení průchodu nulou je jednodušší než detekovat gradient.



Obr. 3.2: Druhá derivace signálu [4]

Ve druhé derivaci se používá funkce Laplacian, která má tu vlastnost, že je stejná ve všech směrech. Laplaceán je tedy invariantní vůči rotaci, platí: [4]

$$\Delta^2 f(j, k) = \frac{\delta^2 f(j, k)}{\delta^2 j} + \frac{\delta^2 f(j, k)}{\delta^2 k}. \quad (3.6)$$

Nevýhoda Laplaceánu je na některé hrany v obraze dvojité odezvy. Podle Marr a Hildrith lze vypočítat Laplaceán Gaussovske funkce takto: [4]

$$H(x, y) = -\Delta^2 \{G(x, \sigma) * G(y, \sigma)\}. \quad (3.7)$$

Tato funkce je aproximovatelná funkcí diferenciálu Gaussovske funkce: [4]

$$H(x, y) = G(x, \sigma_1) * G(y, \sigma_1) - G(x, \sigma_2) * G(y, \sigma_2). \quad (3.8)$$

Jako další nevýhody druhé derivace, která detekuje průchod nulou, je potřeba zmínit velké vyhlazování obrazu, ztráta ostrých rohů a sklon vytvářet uzavřené smyčky hran.

### 3.2.4 Houghova transformace

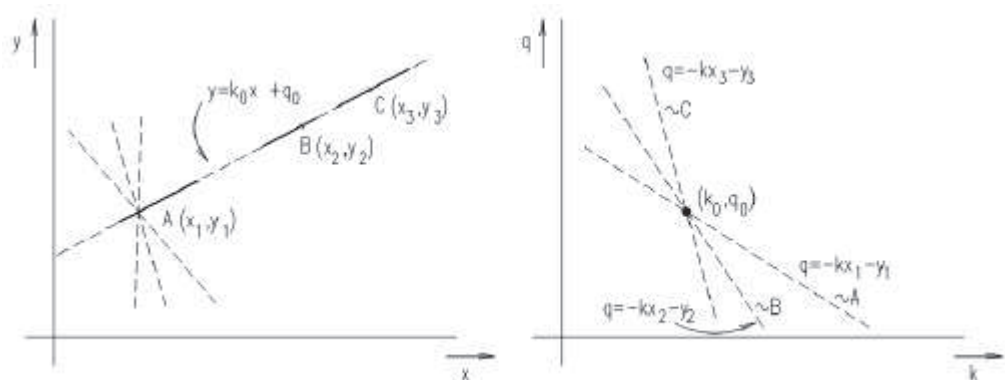
Houghova transformace popisuje hrany s větší přesností, ale je výpočetně mnohem náročnější. Musíme znát analytický popis tvaru hledaného objektu, to znamená detekci známého jednoduchého tvaru – přímku, kružnici, elipsu nebo trojúhelník. Při použití Houghovy transformace dostáváme nejlepší výsledky při použití na binární (naprahovaný) vyhranovaný snímek. Velkou výhodou je, že segmentace není příliš citlivá na porušená data a šum a je použitelná i při částečném zakrytí objektu. Jedná se o transformaci z Kartézského souřadnicového systému do polárního [1].

Parametrické vyjádření přímky: [4]

$$\rho = x * \cos\theta + y * \sin\theta, \quad (3.9)$$

kde  $\rho$  je normalizovaná vzdálenost přímky od počátku a  $\theta$  je úhel vzhledem k ose x.

Křivky v parametrickém obrazu tvoří body obrazu. Jestliže uvažujeme body z jedné přímky, pak křivky se s velkou pravděpodobností protínají v bodě reprezentující parametry hledané přímky.



Obr. 3.3: Houghova transformace přímky[6]

Problém v přesnosti se jeví jako nevýhoda této metody. Tato nevýhoda spočívá v určení jen jednoho maxima pro blízké rovnoběžné čáry vlivem diskretizace. Může nastat také zkreslení, které „zakřiví“ přímky, tím vznikne v parametrickém prostoru více maxim, které představují několik přímek. Metoda také neříká nic o počátku nebo konci křivek a tím získáváme například přímky místo úseček. Pokud použijeme hodnotu  $\rho$  již v průběhu detekce hran, docílíme urychlení algoritmu [5].

Parametrické vyjádření kružnice, podle [5]:

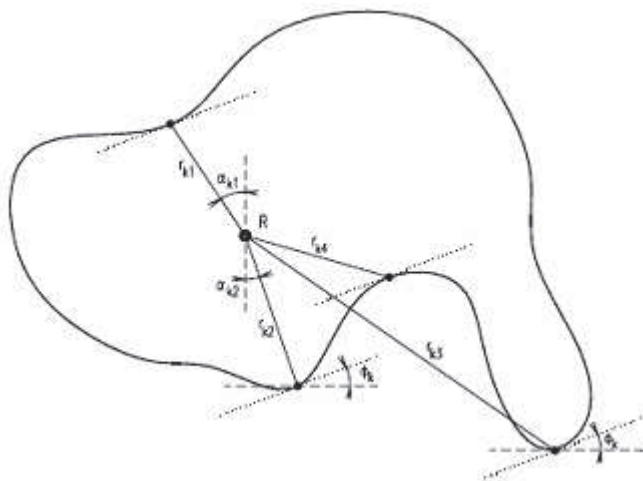
$$x = a + r * \cos\alpha, \quad (3.10)$$

$$y = b + r * \sin\alpha, \quad (3.11)$$

parametry  $a$ ,  $b$  a  $r$  jsou parametry, které hledáme. Definiují střed kružnice ( $a$ ,  $b$ ) a její poloměr ( $r$ ). Z toho plyne, že výpočet souřadnic bude složitější, protože Houghův prostor má tři dimenze. Pro usnadnění práce a rychlejší výpočet souřadnic nám postačí znát nebo odhadnout jeden parametr kružnice.

Stejný princip můžeme použít i pro křivky, elipsy a pro další obrazce, které můžeme popsat parametricky.[2]

Takzvanou Obecnou Houghovu transformaci můžeme použít i na detekci tvarů, které nejdou přímo popsat parametricky. Pomocí explicitního seznamu všech bodů hranice získáme pozice všech pixelů vztahených k referenčnímu bodu – například k těžišti [1].



Obr. 3.4: Ukázka popisu hranice [6]

### 3.3 Regionově orientovaná segmentace

Pro zašuměné obrazy se nehodí hranové detektory, neboť špatně detekují hrany. Pro tyto obrazy je vhodnější metody vycházející z regionově orientované segmentace. Tato metoda vyhledává v obraze homogenní oblasti, které mohou být zadány podle různých parametrů, např. úrovní šedi, barvou, texturou, tvarem, modelem atd.

#### 3.3.1 Segmentace růstem oblastí

Tato metoda je známá nejjednodušším výpočtem - sousední pixely s podobně definovaným kritériem homogenity jsou seskupovány do narůstající oblasti a vytvářejí segmentovanou oblast. Při segmentaci zvolíme na obraze pixel ( $p_s$ ), který bude pro segmentaci výchozím bodem. Tento označený pixel se inicializuje interaktivně, stochasticky nebo podle předchozí analýzy. Po inicializaci se začne zkoumat homogenita sousedních pixelů. Rozhodnutí o přidání pixelu ( $p_j$ ) se rozhoduje pomocí daného kritéria. Tyto kritéria rozdělujeme na statické a dynamické.

Statistické kritérium pro skalární parametr může vypadat podle [1]:

$$|p_s - p_j| \leq T, \quad (3.12)$$

kde  $T$  je zvolené kritérium pro porovnání přidání okolních pixelů  $p_j$  k označenému pixelu  $p_s$ .

Dynamicky proměnné kritérium dovoluje postupnou změnu vstupních parametrů. To má za následek narůstání „pomalu“ proměnné oblasti. Náhlá změna parametrů způsobí zastavení nárůstu a určí hranici oblasti.

Dynamické kritérium vůči právě přidanému pixelu ( $p_i$ ), podle [1]:

$$|p_i - p_j| \leq T, \quad (3.13)$$

Dynamické kritérium vůči aktuální střední hodnotě ( $p_a$ ), podle [1]:

$$|p_i - p_a| \leq T. \quad (3.14)$$

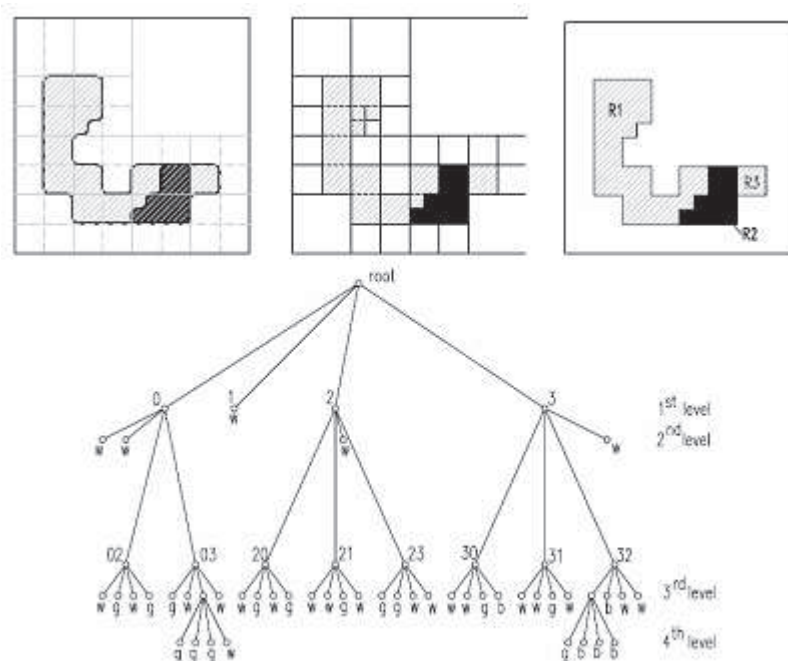
Kritéria pro vektorový prostor lze definovat obdobně. Zvolená tolerance, která rozhoduje o nárůstu oblasti, se vztahuje např. k vzdálenosti vektorů (příslušnost kulovému okolí) nebo může být i nesymetrická tolerance.

Různé výsledky segmentace oblasti lze dosáhnout definicí počátečního pixelu, různě zvolených kritérií, pořadím předkládaných oblastí a počátky zpracování pixelů.

#### 3.3.2 Segmentace dělením a slučováním oblastí

Základní princip metody lze odvodit z jeho názvu. Základem postupu je rozdělení obrazu na překrývající se oblasti, které jsou následně porovnávány. Jestli jsou homogenní, potom se sloučí. Pokud nejsou homogenní zase se rozdělí už rozdělená část obrazu na další nepřekrývající se oblasti. Ty jsou zase porovnávány, slučovány a

děleny. Z tohoto principu vznikne pyramidové rozdělení obrazu, které má úsporný a efektivní popis obrazu. V praxi se osvědčilo dělení obrazu na čtverce, kvůli svojí efektivitě. Takové metodě dělení se říká dekompozice čtyřstromem, protože se popisuje pomocí struktury „čtyřstrom“.



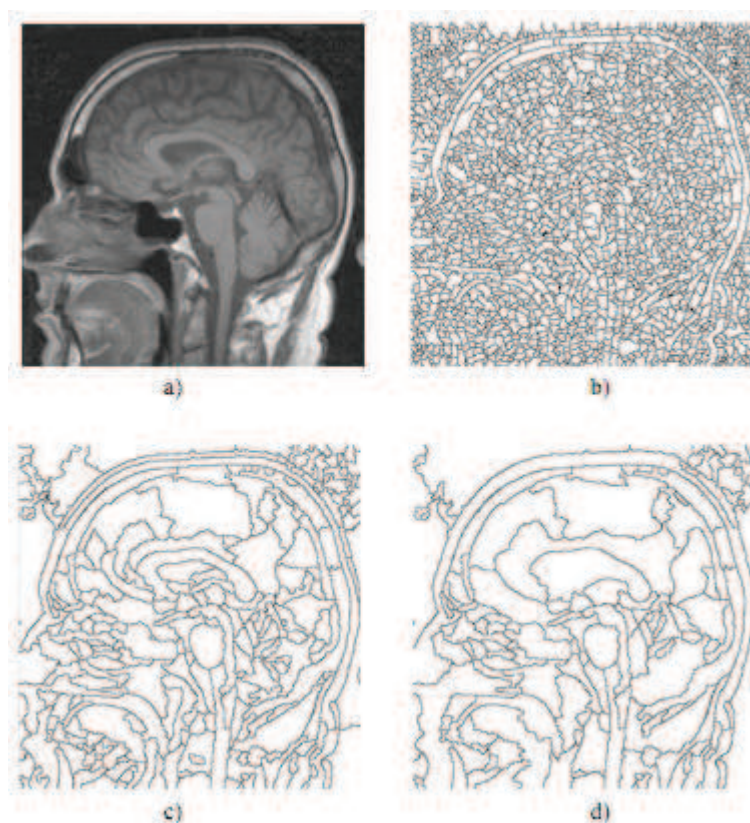
Obr. 3.5: Dekompozice čtyřstromem [6]

Obraz je kořen stromu. Ten se dělí na čtyři kvadranty, u kterých se zjišťuje podmínka stejnorodosti. Pokud určitá oblast podmínku splňuje, znamená to, že se už dále nedělí. Pokud oblast podmínku nesplňuje (např. oblast 0 na obrázku 3.5), musí se rozdělit opět na čtyři části. Tento proces se opakuje do té doby, dokud je co dělit. Dělení však můžeme ukončit i dříve, pokud byla vytvořena podmínka, aby hloubka stromu nepřekročila určitou mez. Pokud tomu tak není, je dekompozice ukončena až nalezením všech čtverců splňujících podmínku. Na závěr nám vzniknou bloky různých velikostí. Informace o blocích nalezneme v uzlech stromu. [1]

### 3.3.3 Segmentace Watershed

Tato metoda je založena na představě rozlévání vody na nerovnoměrný povrch, odtud vychází i její název „rozvodí“. Obraz je chápán jako terén, který je postupně zaplavován vodou. Metoda se nejdříve zaměřuje na nejnížší body nížin. Hladina vody stoupá, v místech, kde by se voda ze dvou různých povodí mohla slít, jsou vytvořeny hráze. Postupné zaplávání obrazu je zastaveno ve chvíli, kdy dosáhneme nejvyššího bodu v terénu, to znamená, že nám vznikne jedna vodní plocha, která obsahuje celou oblast. [2]



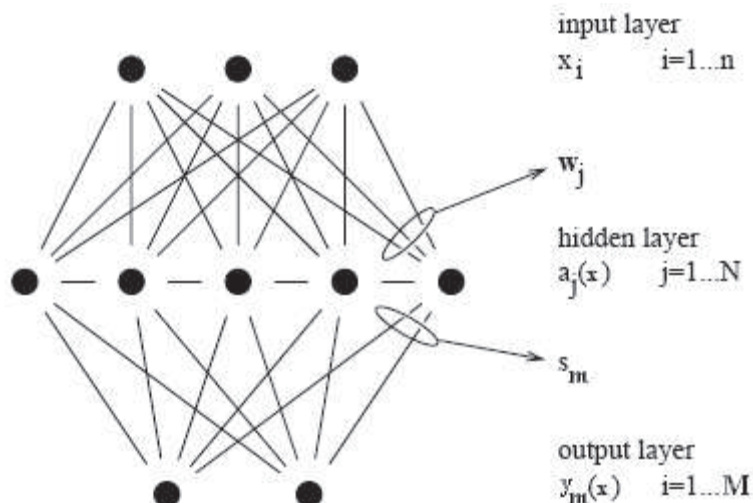


Obr. 3.6: Originální obraz (a) a ukázky segmentace algoritmem watershed s různým rozlišením úrovní (b, c, d) [6]

### 3.3.4 Neuronové sítě

Většina metod segmentace obrazu je založena na znalostech a zkušenosti, jak a podle čeho by měla probíhat. Opakem je segmentace obrazu neuronovými sítěmi, která není založena na podobných meta-pravidlech. Trénování neuronové sítě (dále jen *NN* - *Neural Network*) čistě orientované na data probíhá podle principu "učení příklady". Na druhé straně leží algoritmy analyzující data vzhledem k zadané množině pravidel a příznaků.[10]

Pro trénování umělé inteligence existují dvě metody. První metoda porovnává charakteristické vlastnosti vstupních dat a rozděluje je do skupin bez jakékoliv další interpretace. Tato metoda se nazývá učení bez učitele. Vychází s postupně probrání celé tréninkové množiny a po uvedení jednoho tréninkového typu se docílí ke kompetici neboli soutěži neuronů. Po vyhodnocení se upravují váhy neuronů v okolí a i vítězný neuron. To má za následek vylepšení pozice vítězného neuronu a jeho okolí vůči nové tréninkové sekvenci. Druhá metoda potřebuje ručně zadaná segmentovaná data trénovací sekvence a nazývá se trénování s učitelem. Vstupem této metody jsou příznakové vektory a funkce, která přiřadí vstupnímu vektoru určitý segment obrazu. Kombinací obou dvou metod se spojují výhody jednotlivých metod, vznikla např. *GRBF* neuronové sítě.



Obr. 3.7: Struktura neuronové sítě s radiální bázovou funkcí [10]

Architektura GRBF sítě je třívrstvá (viz. Obr. 3.7). Skládá se ze vstupní vrstvy, jedné skryté vrstvy a výstupní vrstvy. Propojení neuronů mezi vrstvami je úplné. Každý neuron skryté vrstvy je spojen se všemi neurony vstupní i výstupní vrstvy pomocí synaptických vah. Vstupní vrstva neuronů slouží pouze k přiložení klasifikovaného příznakového vektoru  $x$  na vstup sítě. Hodnoty přiloženého vektoru propaguje do skryté vrstvy, kde jsou použity neurony s radiální bázovou funkcí.[11]

### 3.4 Segmentace metodou pružných a aktivních kontur

Metoda založena na optimalizaci úprav hrubých hranic segmentované oblasti podle obsahu skutečného obrazu. Během segmentace předchází obraz příliš členitým hranicím v důsledku šumu. Kontury dělíme na parametrické pružné, geometricky pružné a aktivní kontury.

#### 3.4.1 Parametricky pružné kontury

Podstata metody závisí na volbě počáteční křivky, zadané parametry funkce  $X, Y$  a parametru  $p_0$ . Parametr  $p_0$  je vektor ovlivňující skutečný tvar křivky. Pro nalezení žádaného segmentovaného objektu se optimalizuje tvar hranice podle vektoru  $p_{opt}$ . Tento vektor deformuje tvar hranice objektu na základě obsahu obrazu.

Parametrická kontura je dána vztahem, podle [6]:

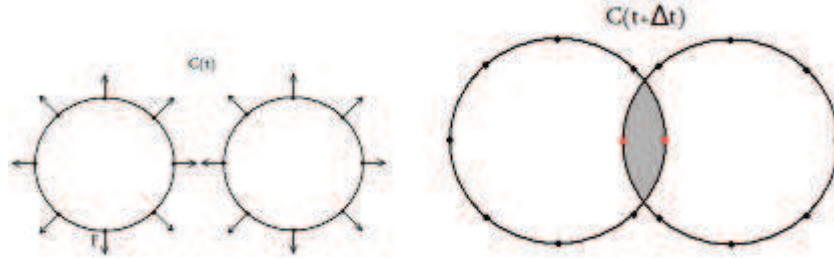
$$X(s, p) = [X(s, p), Y(s, p)]^T, s \in <0, 1> \quad (3.15)$$

kde  $s$  je skalární nezávisle proměnný parametr.

Mezi jednu z největších nevýhod parametrického přístupu patří nemožnost rozšíření do více dimenzí. Řada metod dává totiž uspokojivé výsledky pouze ve dvojrozměrném prostoru, ale problém nastává při rozšíření do třech dimenzí. Důvodem je parametrické vyjádření a použití rovnice pro pohyb rozhraní, protože rozšíření do třetí dimenze znamená nárůst složitosti implementace, která není jednoduchá. [2]



Další potíže nastanou při změně topologie hranice. Při pohybu se oddělené hranice určitých oblastí mohou spojovat, rozdělovat, nebo dokonce zanikat. Proto musíme při výpočtu používat složité postupy, které tyto hranice určí a vyhodnotí [12].



Obr. 3.8: Spojování hranic [12]

### 3.4.2 Metoda geometrických kontur

Tato metoda je definovaná pomocí vrstevnic proměnného skalárního pole  $\Phi(x, y, t)$   $\{(x, y): \Phi(x, y, t) = a\}$ , obvyklá je volba  $a = 0$ . Podle [10] má kontura tvar:

$$X(s, t) = [X(s, t), Y(s, t)]^T. \quad (3.16)$$

Volba funkce  $\Phi(x, y, t=0)$  pro počáteční tvar křivky se obvykle volí jako vzdálenostní mapa k nejbližšímu bodu křivky:  $\Phi(x, y, 0) = \text{dist}(x, y)$ . Vývoj křivky  $\Phi(x, y, t)$  pro  $t > 0$  popisují následující vzorce, podle [10]:

Funkce popisující křivku:

$$\Phi(x, y, t) = \Phi(X(s, t), t) = 0, \quad (3.17)$$

diferenciální rovnice:

$$\frac{\partial \Phi(X(s, t), t)}{\partial t} + (\nabla \Phi(X(s, t), t))^T \frac{\partial X(s, t)}{\partial t} = 0, \quad (3.18)$$

kde normálový vektor  $N$  ke křivce je dán gradientem:

$$N = -\nabla \Phi. \quad (3.19)$$

Rovnice popisující vývoj křivky tvaru je dána pohybem kontury kolmo k tečně. Tedy vlastně ve směru normálového vektoru  $n$ . Tato rovnice se nazývá diferenciální rovnici vývoje [10]:

$$\frac{\partial \Phi(x, y, t)}{\partial t} = \frac{\kappa(x, y, t) + V_0}{|\nabla(G_a * I)(x, y)| + 1} |\nabla \Phi(x, y, t)|, \quad (3.20)$$

kde  $\kappa$  je křivost křivky  $X$ , ta má za následek vyhlazování křivky,  $V_0$  je konstanta „tlakové“ tendence, která udává podle znaménka směr pohybu křivky buď smršťování nebo expanzi. Jmenovatel  $|\nabla(G_a * I)(x, y)| + 1$  je zastavující faktor, snižující rychlost v oblastech abs. vysokého gradientu parametrického obrazu odvozeného

z analyzovaného obrazu I, až do zastavení [10].

Při segmentaci se postupně zvolí počáteční parametry kontury, provede se počáteční výpočet pomocí  $\Phi(x, y, t=0)$  a nakonec se zkoumá časový vývoj pomocí parciální diferenciální rovnice 5.6. Nesporné výhody metody spočívají změnu topologie hranic, tzn. slučování uzavřených křivek nebo také rozdělování. Naopak nevýhoda spočívá v nerozpoznání okrajů při slabých hranicích v obraze, to má za následek nezastavení vývoje.

### 3.4.3 Metoda aktivních kontur

Aktivní kontury je metoda postupného tvarování kontur až ke hraně objektu v obraze. Model aktivní kontury je řízená uzavřená kontura, která se deformuje vlivem tzv. vnitřních, obrazových a vnějších sil. Vnitřní síly kontrolují hladkost průběhu, obrazové síly směřují tvarování kontury směrem ke hraně objektu a vnější síly jsou výsledkem počátečního umístění kontury.[4]

Výpočet konečné aktivní kontury v pomocí těchto sil závisí na lokálním minimu energie kontury, podle [4]:

$$E_s = \sum_{n=1}^N E_N\{v\} + \sum_{n=1}^N E_I\{v\} + \sum_{n=1}^N E_T\{v\}, \quad (3.21)$$

kde  $E_N$  reprezentuje vnitřní energii kontury  $v$ ,  $E_I$  představuje energii obrazu a  $E_T$  je energie omlazení. Vnitřní energie kontury řídí ohyb a zlomy kontury  $v$ .



Obr. 3.9: Postup algoritmu aktivní kontury[4]

Při složitějších konturách přijatelné kvality standardní algoritmus produkuje falešné kontury nebo smyčky, to je způsobeno nereagováním na prudké zlomy hranic objektu. Tyto nedostatky lze odstranit modifikovanou verzí.

## 4 ZÁKLADNÍ STRUKTURA PROGRAMU SEGANAG

K tvorbě aplikace bylo použito prostředí programu Visual studio 2008 a programovací jazyk C++. K usnadnění práce se zpracováním obrazu byla do prostředí implementována volně šiřitelná knihovna openCV. Uživatel ovládá program prostřednictvím grafického rozhraní, které odpovídá klasické oknové aplikaci operačního systému Windows.

### 4.1 Knihovna OpenCV

OpenCV(Open-source Computer Vision) je volně dostupná knihovna pro počítačové vidění. Tato knihovna obsahuje velké množství funkcí pro zpracování obrazu i videa v reálném čase. Historie knihovny se začala psát v lednu roku 1999, kdy byla společností Intel vydána její první alpha verze. V této době už existuje verze openCV 2.3. Všechny verze jsou vydávány s BSD licencí, což znamená volné šíření licencovaného obsahu pod podmínkou uvedení autora a informací o licenci.

Jedním z klíčových cílů OpenCV je vytvořit jednoduše použitelnou infrastrukturu algoritmů počítačového vidění. Knihovna nyní obsahuje více jak 500 funkcí z mnoha odvětví počítačového vidění. OpenCV taktéž zahrnuje knihovnu pro strojové učení (Machine Learning Library), jelikož ta je s počítačovým viděním úzce spjata. [15]

### 4.2 Vývojový diagram

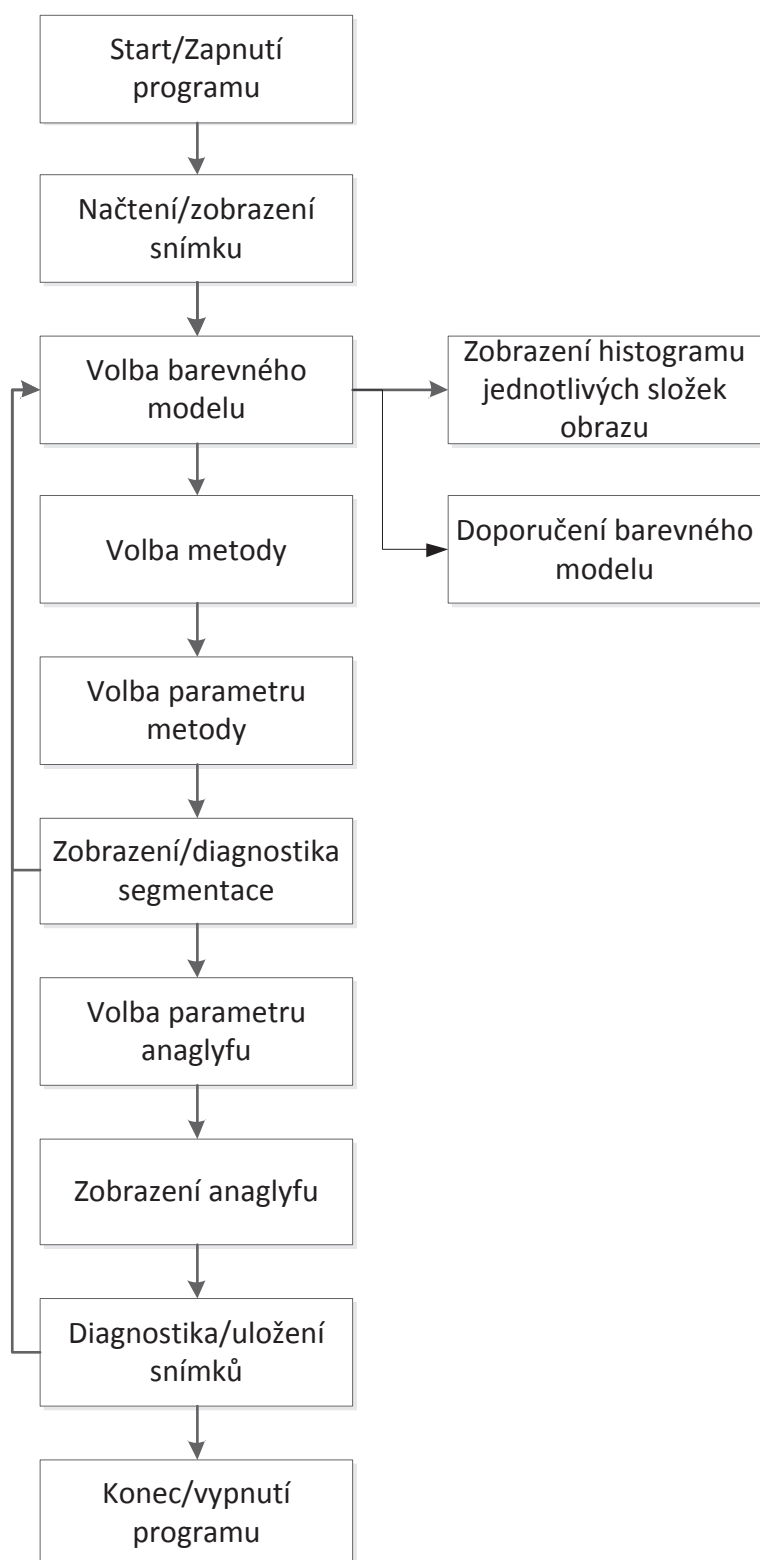
Postup práce s programem popisuje vývojový diagram zobrazen na obr. 4.1. Diagram zobrazuje jednotlivý postup uživatele při práci s programem. Neznázorňuje samostatné činnosti algoritmů programu. Tyto algoritmy jsou popsány níže v podkapitolách.

Po spuštění programu SegAnag a načtení snímku, s kterým chce uživatel nadále pracovat, je automaticky zobrazen a jeho data jsou nahrány do struktury IplImage k dalšímu zpracování. V dalším kroku může uživatel provést transformaci snímku, který je původně popsán v modelu RGB, do jednoho z dostupných modelů: HSV, XYZ, YCrCb, Lab, Luv, HLS. Poté si může nechat zobrazit histogramy jednotlivých obrazových složek převedeného snímku.

Na základě zobrazení histogramů může uživatel zvolit barevný model vstupního snímku. Jestli se uživatel nemůže rozhodnout, který model zvolí, může použít doporučení. To doporučí vhodný barevný model. Následně uživatel zvolí metodu segmentace a vstupní parametry. Po zobrazení výsledků segmentace uživatel subjektivně zhodnotí výsledek. Při nedostatečném výsledku segmentace může uživatel změnit používaný model, použitou metodu segmentace či její parametry.

Poté si může nechat zobrazit anaglyfický 3D snímek, kdy stačí zadat parametr posunutí, na kterém závisí velikost prostorového efektu. Dále má uživatel několik možností. Jestliže je uživatel spokojen může snímky, jak výsledek anaglyfu, tak

výsledek segmentace, uložit na pevný disk. Nebo změnit parametry anaglyfu. Nakonec se může celý proces opakovat pro jiný snímek nebo aplikaci ukončit.



Obr. 4.1: Vývojový diagram

## 4.3 Grafické rozhraní

Grafické rozhraní se skládá pouze z hlavního okna realizovaného pomocí komponenty *Form*, který slouží jako pracovní plocha pro umístění zbylých ovládacích prvků.

Pro zobrazení veškerých obrazových dat jak vstupních, tak výstupních obrázků (převedeného na barevný model, výstup segmentace), slouží *PictureBox*. V něm byla použita vlastnost *Image* na vykreslení dat. Další funkcí *PictureBoxu* bylo použito *SizeMode* pro přizpůsobení snímku velikosti okna.

Většina algoritmů reaguje na ovládací prvek *button* tedy tlačítko a její událost *Click*. Pro výběr barevného modelu a volby segmentace byla svěřena komponentě *ComboBox*, kde využíváme vlastnosti *SelectedIndex*. Tento prvek vytváří rozbalovací menu sloužící k výběru požadovaných činností.

Pro zadávání parametrů segmentace byly použity komponenty s názvem *NumericUpDown*, kde se nastavuje číslo. U ní se využilo vlastnosti *maximum* a *minimum* na nastavení intervalu mezi vstupních parametrů. Pro přečtení hodnoty nastavenou v prvku slouží vlastnost *value*. Pro textový popis komponent byly použity komponenty *label*. Pro výběr vstupního bodu u segmentační metody růst ze semínka byl použit *checkBox*, u kterého jsme využili vlastnosti *checked*. Této vlastnosti bylo použito i u uložení snímku, kdy vybíráme výsledek k uložení.

Pro průvodce programem slouží prvek *RichTextBox*, do kterého vkládáme text pomocí vlastnosti *Text*.

Neméně důležité jsou také nezobrazované komponenty v grafickém prostředí, ale bez nich by program nemohl správně fungovat. Je to hlavně komponenta k načtení souboru, v našem případě obrázku, *OpenFileDialog*, k uložení souboru (obrázku) *SaveFileDialog* a komponenta *ContextMenuStrip* nabízející po stisku sekundárního tlačítka myši nabídku na další funkce programu.

## 4.4 Popis části aplikace

Aplikace se skládá z různých částí, které dohromady tvoří celou aplikaci. V této části si popíšeme, jak fungují jednotlivé části programu. Jelikož bylo použito knihovny openCV jsou věci s prací s obrazem ulehčeny o některé funkce.

### 4.4.1 Načtení, zobrazení a převedení snímku na barevný model

Načtení souboru obrazového typu probíhá přes funkci *OpenFileDialog*, která nám vyvolá okno pro výběr souboru. Ve filtru funkce je přidána podmínka na výběr obrázků, která je řešená filtrací přípon souborů. Ty musí mít koncovky *.jpg*, *.bmp*, *.tiff*, *.png*. Soubor se následně uloží do formátu knihovny openCV *IplImage*, s kterým potom pracujeme po celou dobu, přes příkaz *cvLoadImage*. Tato funkce vyžaduje vstupní parametry cestu k souboru a jeho název a v jakých barvách má být reprezentován. Než je tak učiněno musí se cesta uložená z typu *string* z funkce *openFileDialog* převedena na typ *char\** pomocí funkce převodu *StringToHGlobalAnsi*.

Pro zobrazení snímku v *pictureBoxu* musíme vyčíst obrazová data z *IplImage* do

předem vytvořené Bitmapy se stejnými rozměry jako je snímek nahraný v `IplImage`. Po té už pak převedený snímek vykreslíme do `pictureBoxu`.

Pro převod na barevný model byly použity funkce knihovny openCV `cvCvtColor`, kde se zadávají parametry vstupní a výstupní obraz a parametr na jaký chceme snímek převést. Tento parametr má tvar `CV_XXX2YYY`, kde `XXX` doplníme barevný model reprezentující vstupní snímek, v našem případě RGB, a kde za `YYY` doplníme model pro převedený snímek. Pro převod stejných barevných modelu, které tato funkce nepodporuje, je použito funkce `cvCopyImage`, která zkopíruje vstupní snímek do výstupního bez jakékoliv změny reprezentace obrazových bodů.

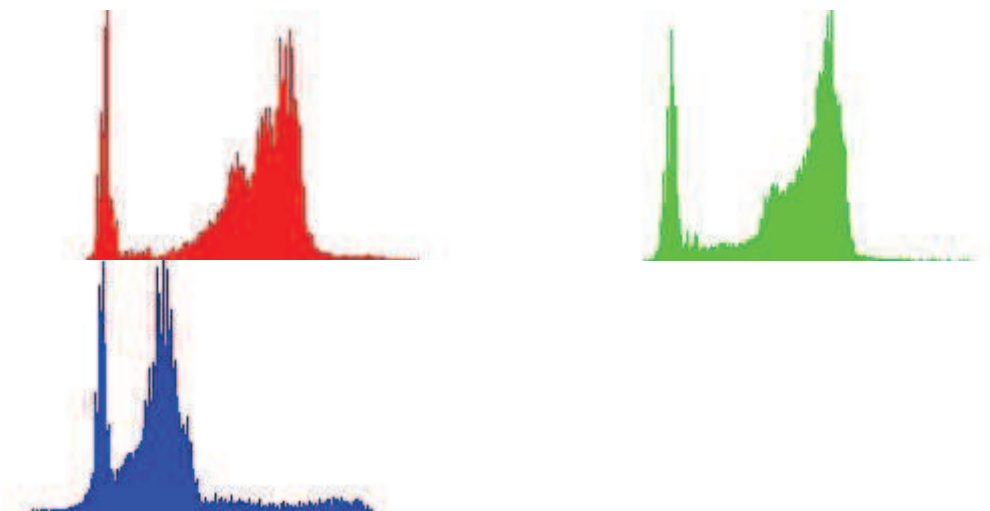
#### 4.4.2 Zobrazení histogramu složek

Pro zobrazení histogramu jednotlivých obrazových složek barevného modelu slouží funkce:

```
void show_histogram(char* popis_okna, IplImage* src, char*  
nazev_kanalů),
```

kde vstupem funkce je název nového okna, ve kterém se histogram zobrazí, snímek, z kterého se histogram spočítá a název kanálu odpovídající barevné složce obrazu, která se má zobrazit.

Jádrem této funkce je cyklus, který prochází jednotlivé body reprezentující danou složku v obraze a inkrementuje příslušnou hodnotu v reprezentující pole všech hodnot ve snímku. Poté zjistí maximální hodnotu z histogramu, nanormuje celý histogram pro zobrazení snímku v dané velikosti zobrazovaného okna. Funkce tedy vrací obrázek histogramu zbarvený podle kanálu barvami modelu rgb, tedy pro první kanál barevného modelu je zobrazen červenou barvou, pro druhý kanál zelenou barvou a pro třetí kanál barvou modrou (viz Obr. 4.2).



Obr. 4.2: Histogram obrazových složek R (vlevo nahoře), složky G (vpravo nahoře) a složky B (vpravo dole)

#### 4.4.3 Funkce určující vhodný model pro segmentaci

Tento algoritmus vychází z vývojového diagramu z obrázku 4.3. Jako hlavní náplní je doporučit ve kterém barevném modelu se má segmentace provádět a jaký procentní

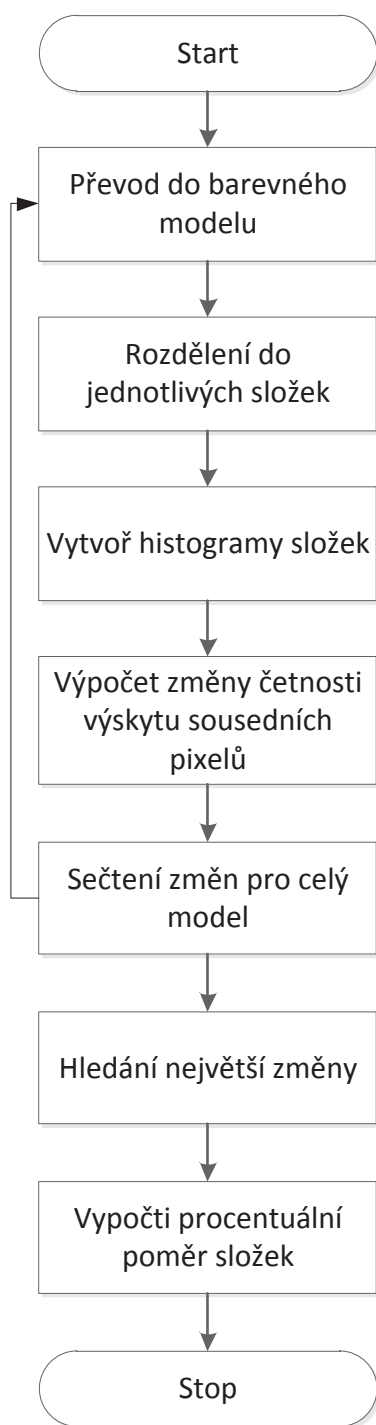
vliv mají jeho složky na složení obrazu.

Na začátku algoritmu je vstupem snímek nahraný v RGB modelu. Po té je postupně převáděn na další dostupné modely: HSV, XYZ, YCrCb, Lab, Luv, HLS. Následuje krok, ve kterém rozdělíme převedený snímek na jednotlivé obrazové složky pomocí funkce *cvSplit*. Poté následuje pro každou takto rozdělenou složku zavolat funkci:

```
int Dop_bar_schem(IplImage *inimg),
```

ve které se nejprve z dané obrazové složky vytvoří histogram, reprezentující četnost výskytu pixelů dané velikosti. Dále se v cyklu prochází celý histogram a vypočítává se součet rozdílů sousedních četností vztahený k maximální hodnotě v histogramu.

Tyto výsledné součty pro celý obrazový model jsou následně sečteny dohromady a pomocí podmínky se hledá největší hodnota změn obrazového modelu. Následně je vybrán tento model a vypočítává se procentuální poměr složek reprezentující tento model.



Obr. 4.3: Vývojový diagram funkce doporučení

#### 4.4.4 Segmentační metoda využívající metodu Kmeans

Jako inspiraci jsem použil open source skript z matlabu dostupný na <http://www.mathworks.com/matlabcentral/fileexchange/25257-color-image->



segmentation.

Tato metoda vychází z algoritmu k-means. Jeho vstupem je množina dat a číslo  $k$  udávající počet tříd, do kterých se postupně rozdělují vstupní data. V openCV je funkce *cvKMeans2*, do které zadáváme vstupní matici bodu, počet tříd, výstupní data rozdělené do tříd a počet maximálních iterací.

Po zadání parametrů se začnou iterativně opakovat následující kroky:

1. Klasifikace: Všechna data  $x_i$ ,  $i = 1, \dots, l$  se klasifikují do tříd určených vektory  $\mu_j$ ,  $j = 1, \dots, k$  podle minima euklidovské vzdálenosti. Tedy vzor  $x_i$  je přiřazen do třídy  $y_i$  podle [16]:

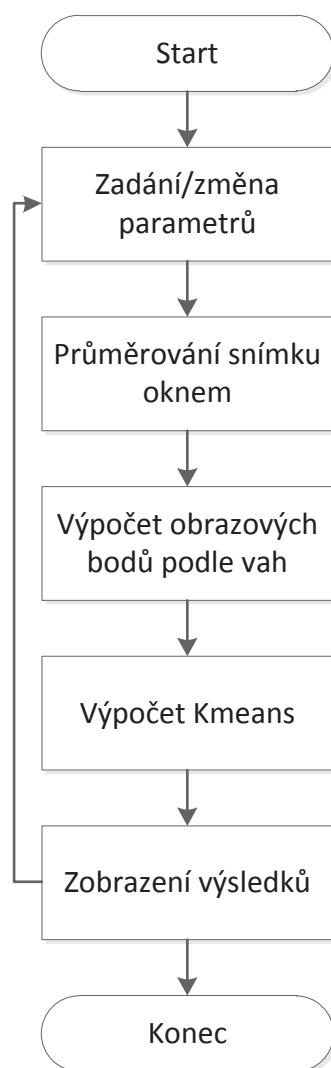
$$y_i = \operatorname{argmin}_{j=1,\dots,k} \|x_i - \mu_j\|. \quad (4.1)$$

2. Přepočítání vektorů  $\mu_j$ : Vypočítají se nové hodnoty vektorů  $\mu_j$  jako střední hodnoty dat  $x_i$ , které byly klasifikovány do třídy určené příslušným vektorem  $\mu_j$ . Tedy nová hodnota  $\mu_j$  se spočte podle vztahu [16]:

$$\mu_j = \frac{1}{l_j} \sum_{\substack{i=1 \\ y_i=j}}^l x_i, \quad (4.2)$$

kde  $l_j$  je počet vzorů  $x_i$  klasifikovaných v druhém kroku do třídy určené vektorem  $\mu_j$ . Kroky 1 a 2 se opakují doté doby pokud se alespoň jeden vektor  $x_i$  klasifikuje do jiné třídy, než byl klasifikován v předcházejícím kroku. [16]

U této segmentační metody používáme algoritmus podle vývojového diagramu na Obr. 4.3. Jako vstup je použit snímek převedený do různého barevného modelu. Dále se zadávají parametry velikosti o počtu shluků, průměrovacího okna, váhy jednotlivých složek obrazu. Po zadání parametrů procházíme snímek, který průměrujeme oknem velikosti zadaným v úvodu a vypočítáváme obrazové body podle zadaných vah obrazových složek. Po výpočtu musíme převést obrázek na matici, protože funkce *cvKmeans2* pracuje jen s jednorozměrnou maticí. Potom vypočítáme paletu barev pro všechny shluky jako průměrnou hodnotu všech pixelů co náleží danému shluku. Tyto hodnoty pak nahráváme do výstupního obrázku a následně zobrazíme výsledek. Po zhodnocení snímku můžeme pozměnit data a opakovat celý cyklus znova.



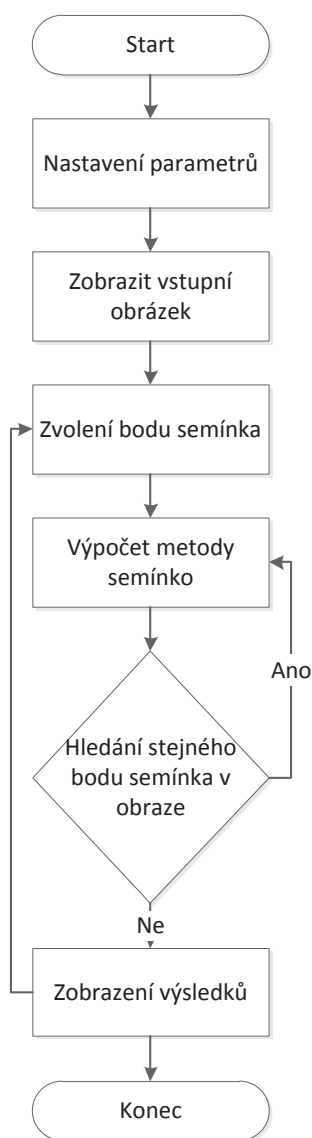
Obr. 4.3: Vývojový diagram metody kmeans

#### 4.4.5 Segmentační metoda růstu ze semínka

Tato metoda vychází s funkce *cvFloodFill*, kdy vstupní parametry jsou vstupní obrázek, bod, ze kterého se začínají hledat podobné pixely tvořící segment, volba barvy, kterou má být nalezené segmenty obarveny, horní a dolní mez podobnosti pixelu a nakonec volba okolí z kterého se počítá podobnost pixelu.

Algoritmus metody je vyjádřen vývojovým diagram na Obrázku 4.4. Vstupem funkce je snímek ve zvoleném barevném prostoru. Nastavíme vstupní parametry, mezi které patří meze podobnosti sousedních pixelů a volba, jestli budeme zadávat vstupní bod do segmentovaného objektu nebo jeho okolí, kdy je toto okolí odstraněno ze snímku. Potom se zobrazí vstupní snímek v novém okně reprezentovaný barevným modelem RGB. Po zobrazení zadáme vstupní bod pro metodu růstu ze semínka myší na obraz. Pomocí funkce *cvSetMouseCallback*, detekujeme stisk tlačítek myši v určeném

okně. Po stisknutí levého tlačítka myši aplikujeme funkci *cvFloodFill*, po výpočtu pro tento bod, se začnou hledat ve vstupním obraze reprezentovaném ve zvoleném barevném modelu totožné pixely, se stejnými velikostmi obrazových složek, a provede se výpočet pro tyto body. Následně se zobrazí výstupní obraz s vysegmentovanou částí. Pro přidání ještě dalšího segmentu k již vysegmentovanému snímku je možnost zadání nového vstupního bodu levým tlačítkem myši na další bod. Tím se spustí se celý cyklus znovu pro novou hodnotu zadaného bodu. Pro vymazání zobrazených segmentů slouží pravé tlačítko myši, kdy se přes funkci *cvCloneImage* nahraje znovu vstupní snímek v RGB.



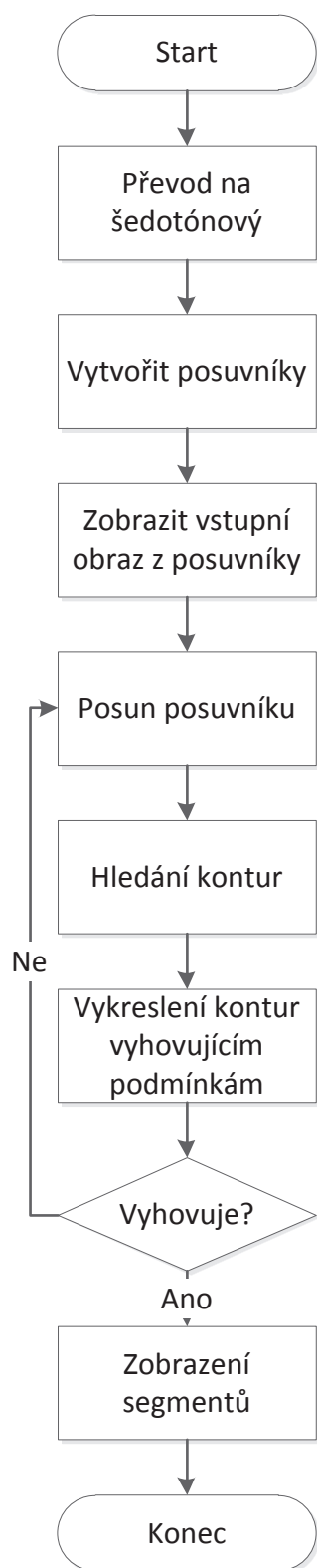
Obr. 4.4: Vývojový diagram segmentační metody semínko

#### 4.4.6 Segmentační metoda kontur

Následující metoda je založena na metodě kontur vycházející z vývojového

diagramu na obrázku 4.5. Jako jediný nepracuje s barevnými modely, ale soustředí se na zpracování v šedotónové oblasti.

Vstupem je nahraný snímek, který se hned převede do šedotónové oblasti pomocí funkce *cvCvtColor*. Dále se definuje okno pomocí funkce *cvNamedWindow*. V tomto okně se vytvoří dva posuvníky pomocí funkce *cvCreateTrackbar*, které budou sloužit na změnu parametrů, a to velikost prahu a obsahu. Při změně parametrů zavolají funkci na výpočet kontur. Kde je nejprve snímek vyhlazen pomocí funkce *cvSmooth*, pak invertován pomocí funkce *cvNot* a následně provedeno prahování pomocí funkce *cvThreshold* námi zvoleného prahu. Následně najde kontury v obraze pomocí funkce *cvFindContours*. Poté se pomocí cyklu vypočítává obsah kontury pomocí funkce *cvContourArea* a vykreslují se jen ty kontury, které přesáhly zadaný obsah a práh. Pro vykreslení kontur slouží funkce *cvDrawContours*. Výstup je zobrazen do okna ve formě součtu původního a segmentovaného snímku pomocí metody *cvAdd*. Při změně hodnoty posuvníků se postup opakuje. Ukončit segmentaci můžeme stisknutím klávesy, tím se funkcí *cvDestroyAllWindows* zavřou všechny otevřená okna, a zobrazí se nám jen nalezené segmenty ve vstupním obraze.



Obr. 4.5: Vývojový diagram segmentační metody kontur

#### 4.4.7 Anaglyf

Pro vytvoření algoritmu pro anaglyf se vychází ze vstupního načteného obrázku, který procházíme v cyklu a postupně čteme hodnotu jednotlivých pixelů pomocí funkce *cvGet2D* a zapisujeme ji do nového posunutého obrázku o zvolený parametr *k*. Tak vznikne nový snímek, podobný vstupnímu, ale posunutý v horizontálním směru o *k*. Tento snímek bude sloužit jako druhý pro zobrazení anaglyfu. Následně po provedení cyklu rozdělíme jak vstupní snímek tak nově vytvořený na jednotlivé obrazové složky RGB pomocí funkce *cvSplit*. Následně sečteme jejich obrazové složky pomocí funkce *cvMerge* tak, že ze vstupního snímku použijeme červenou složku reprezentující snímek a z nově vytvořeného snímku použijeme modrou a zelenou složku. Následně zobrazíme vytvořený anaglyfický snímek.

#### 4.4.8 Uložení obrázku

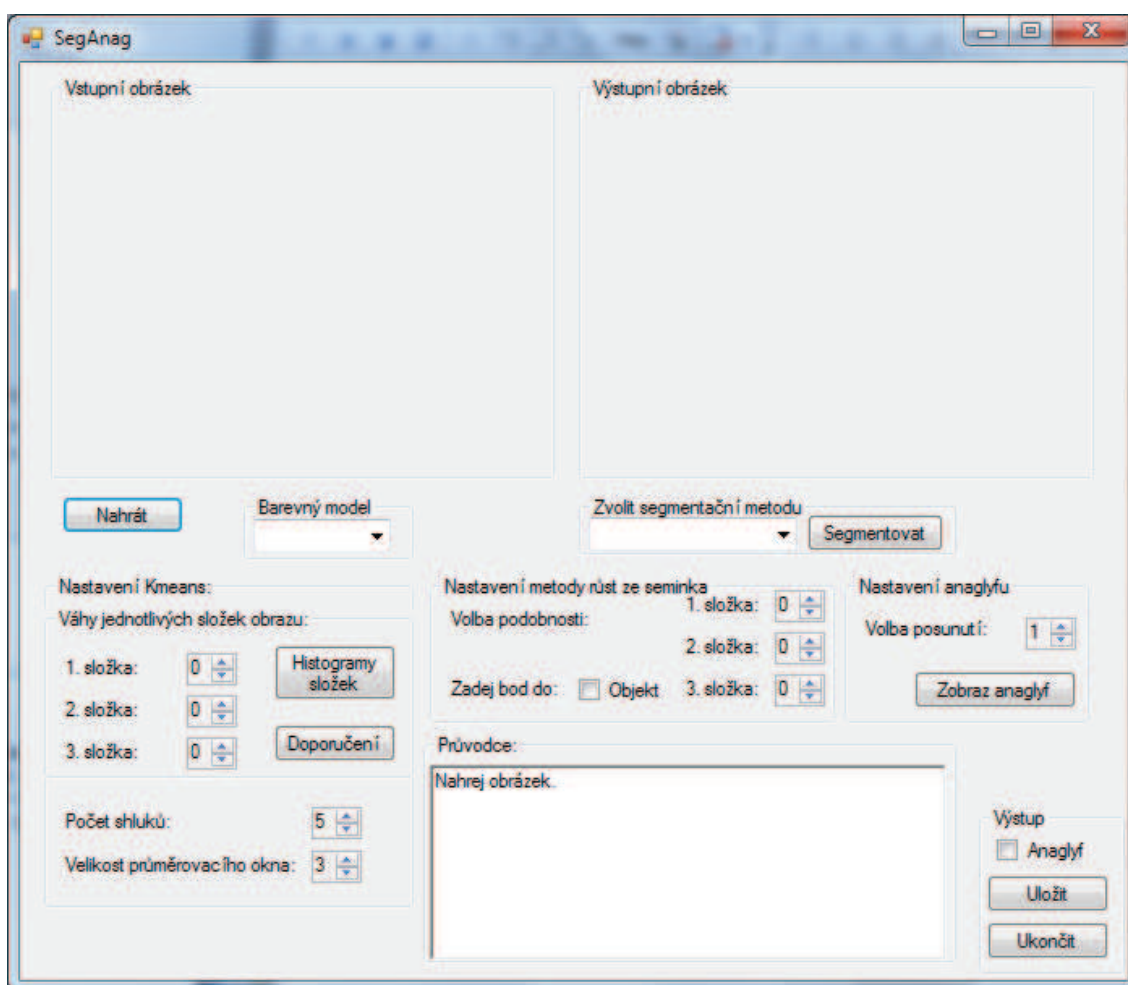
Uložení obrazu se provádí pomocí funkce *cvSaveImage*, do které se zadává cesta s názvem souboru a proměnná reprezentující obrázek ve struktuře *IplImage*. Pomocí dialogového okna vyvolaného při úkonu uložení se zobrazí okno s nabídkou pro uložení snímku ve formátu jpg. Poté musí být cesta s názvem souboru převedena z typu string z funkce *saveFileDialog* převedena na typ *char\** pomocí funkce převodu *StringToHGlobalAnsi*.

## 5 UŽIVATELSKÁ PŘÍRUČKA

Program SegAnag je uživatelsky přívětivý a uživatel k jeho ovládnutí nepotřebuje hrubší teoretické znalosti segmentace ani tvorby anaglyfického snímku. Naopak nabízí možnou změnu vstupních parametrů, která vede k lepším výsledkům segmentace.

### 5.1 Hlavní obrazovka

Po spuštění programu se otevře úvodní obrazovka programu, na které jsou uživateli dostupné všechny funkce programu. V průběhu používání se mu budou zobrazovat výsledky jeho práce.



Obr. 5.1: Obrazovka aplikace SegAnag

Hlavní obrazovka je členěna do několika bloků. V horní polovině okna jsou umístěné dva boxy pro zobrazení snímků. V levém boxu bude vždy znázorněn vstupní nahraný snímek. Kdežto v tom pravém se bude zobrazovat snímek postupně s úkony uživatele. Takže se tam budou zobrazovat snímky po převedení barevného modelu, výsledku segmentace.

Pod boxy zobrazení snímků je umístěno několik bloků pro obsluhu programu. Ovládací prvky jsou následně rozděleny boxy, pro kterou metodu jsou používány. Jediný ovládací prvek, který není přiřazen ani do jednoho boxu je tlačítko Nahrát. Nachází se vlevo uprostřed hlavní obrazovky. To slouží pro načtení vstupního obrazu z disku uživatele a zobrazení v boxu Vstupní obrázek. Uživatel má ještě další možnost jak načíst snímek, a to kliknutím sekundárního tlačítka myši kdekoli na obrazovce programu a výběr Nahrát z objevené nabídky. Dále může ještě z této nabídky vybrat možnosti Czech/English, která přepíná jazyk aplikace s češtiny do angličtiny, a konec, která ukončuje aplikaci. Výběr obrázku probíhá pomocí standardního Windows dialogu. Aplikace může pracovat s obrázky různých formátů (jpeg, bmp, tiff, png).

Dalším blokem následujícím po nahrání obrázku je blok Barevný model, který převede vstupní obraz do jiné reprezentace obrazových dat. Uživatel má na výběr ze široké škály modelů (RGB, HSV, YCrCb, XYZ, Lab, Luv, HLS). Následně je takto převedený snímek zobrazen v bloku Výstupní obrázek, nacházející se v aplikaci vpravo nahoře. Po převedení modelu si uživatel může nechat zobrazit histogramy jednotlivých složek převedeného obrazu. To se zobrazí v samostatných oknech, které na uživatele vyskočí, když zmáčkne tlačítko Histogramy složek, histogramy jednotlivých složek se zavřou stiskem klávesy. Toto tlačítko se nachází už v boxu Nastavení metody Kmeans, protože slouží k rozhodování, které složce dát jakou váhu. V tomto boxu se nachází také komponenty k nastavení váhy jednotlivých složek, počtu shluků a velikosti průměrovacího okna pro vstup segmentace metodou Kmeans. A nakonec se tam také nachází tlačítko Doporučení, které uživateli doporučí, jaké schéma je vhodné použít pro vstupní snímek a také procentuální poměr jednotlivých obrazových složek v obraze. Nicméně tato funkce je doporučující a nezaručuje přesné výsledky.

Následuje blok Zvolit segmentační metodu. Jeho součástí je rozbalovací menu, které slouží k výběru metody segmentace, která se provede po stisknutí tlačítka segmentovat a snímek zobrazí v boxu Výstupní obrázek. Program nabízí tyto metody:

- Kmeans,
- Růst ze semínka,
- Kontury.

U některých metod se uživateli zobrazí okno se vstupním snímkem, kde bude vybírat myší souřadnice vstupního bodu pro metodu růstu ze semínka, nebo se mu zobrazí vstupní obrázek doplněný postníky (Obr. 5.2), kde má možnost měnit nastavení, a tím výsledek segmentace u metody kontur. Jeden posuvník se stará o velikosti hladiny pixelů obrazu a druhým se nastavuje velikost obsahu kontury, která má být zobrazena. Po zavření okna se nahraje snímek do boxu Výstupní obrázek.





Obr. 5.2: Nastavení parametrů u segmentace metodou kontur

Při výběru metody růstu ze semínka musí uživatel zvolit vstupní bod pro výpočet metody. Tento bod zadává, buď v okolí, nebo v objektu. Také může zadat parametry podobností jednotlivých složek, které rozhodují o přidání sousedních pixelů do jednoho segmentu.

Další box s názvem Nastavení anaglyfu se nachází v aplikaci napravo. Uživatel v něm zvolí parametr posunutí a tím rozdíl mezi snímkem reprezentovaným červeným kanálem a snímkem s modrozeleným kanálem. Tlačítko „Zobraz anaglyf“ pak už jen

zobrazí do samostatného okna anaglyfický snímek.

V aplikaci je také umístěn box s názvem Průvodce, ten nás informuje o možnostech aplikace v daném bodě a upozorňuje na chyby, které nám brání provést určitý krok. Nebo při stisku tlačítka doporučení nacházející se v boxu Nastavení metody kmeans nám vypíše radu pro nastavení vstupních parametrů jednotlivých obrazových složek a jaký model má být použit.

V posledním boxu s názvem výstup jsou dvě tlačítka, jedno na ukončení aplikace a druhé s názvem Uložit pro uložení výstupního snímku, kdy zatržením možnosti anaglyf se ukládá anaglyfický snímek, jinak výstup segmentace.

## 6 DOSAŽENÉ VÝSLEDKY

V závěrečné kapitole si ukážeme výsledky dosažené aplikací SegAnag u různých segmentačních metod. U segmentačních metod si ukážeme ještě vliv na nastavení parametrů na výsledek segmentace. Jako vstupní snímek použijeme obrázek 6.1.



Obr. 6.1: Vstupní obrázek

## 6.1 Segmentace založená na metodě Kmeans

U této metody si ukážeme vliv výsledků segmentace na zvolený barevný model, váhu jednotlivých složek barevného modelu, počtu shluků a velikosti průměrovacího okna.



Obr. 6.2: Výsledek segmentace pro barevný model RGB (vlevo) a HLS se stejnou vahou obrazových složek. Počet shluku 5, velikost průměrovacího okna 3.



Obr. 6.3: Výsledek segmentace pro barevný model RGB (vlevo) a HLS (vpravo) s nastavenými vahami 4, 8, 6 pro obrazové složky. Počet shluku 5, velikost průměrovacího okna 3.



Obr. 6.4: Výsledek segmentace pro různý počet shluků, vlevo 3, vpravo 20. Další parametry nastavení konstantní: váhy složek: 4,8,6, velikost okna 3, použitý model HLS.



Obr. 6.5: Výsledek segmentace pro různou velikost průměrovacího okna, vlevo 3, vpravo 8. Další parametry konstantní: váhy složek: 4,8,6, počet shluků: 5, použitý barevný model HLS.



Obr. 6.6: Výsledek segmentace pro nastavení funkce doporučení s barevným modelem HSL a vahami složek: 2,4,3 (vlevo) a pro výsledek segmentace při nastavení barevného modelu HSV s vahami 8,8,4, další parametry: počet shluků 5, velikost průměrovacího okna 5.



Obr. 6.7: Výsledek segmentace pro barevný model YCrCb s vahami složek: 3,2,4, (vlevo) a XYZ s vahami složek: 3,2,3, další parametry: počet shluků 5, šířka průměrovacího okna: 5.

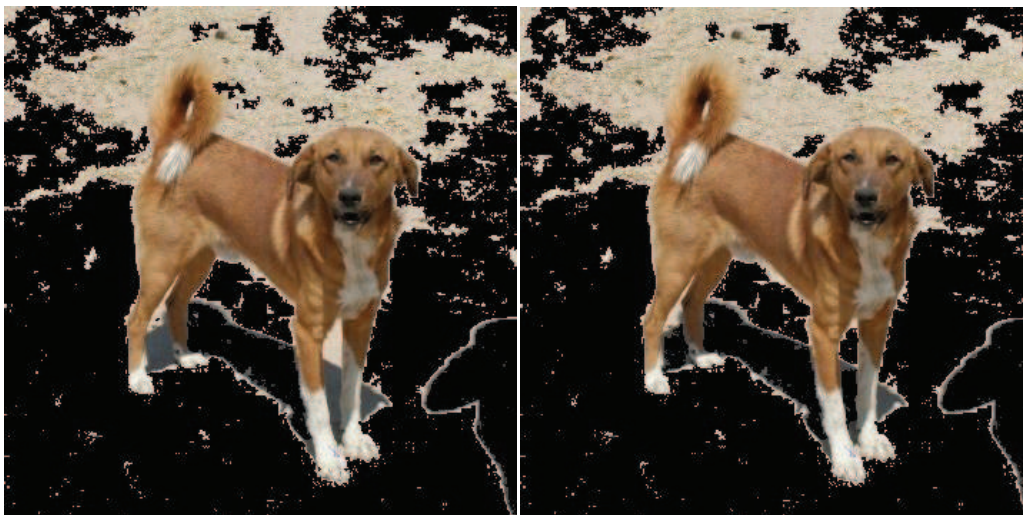




Obr. 6.8: Výsledek segmentace pro barevný model Lab (vlevo) a Luv s parametry: váhy složek: 3,3,3, počet shluků: 5, velikost průměrovacího okna: 5.

## 6.2 Segmentace založená na metodě semínko

U této metody si taktéž znázorníme výsledky na použitý barevný model a vliv zvolení prahu podobnosti se vstupním bodem.



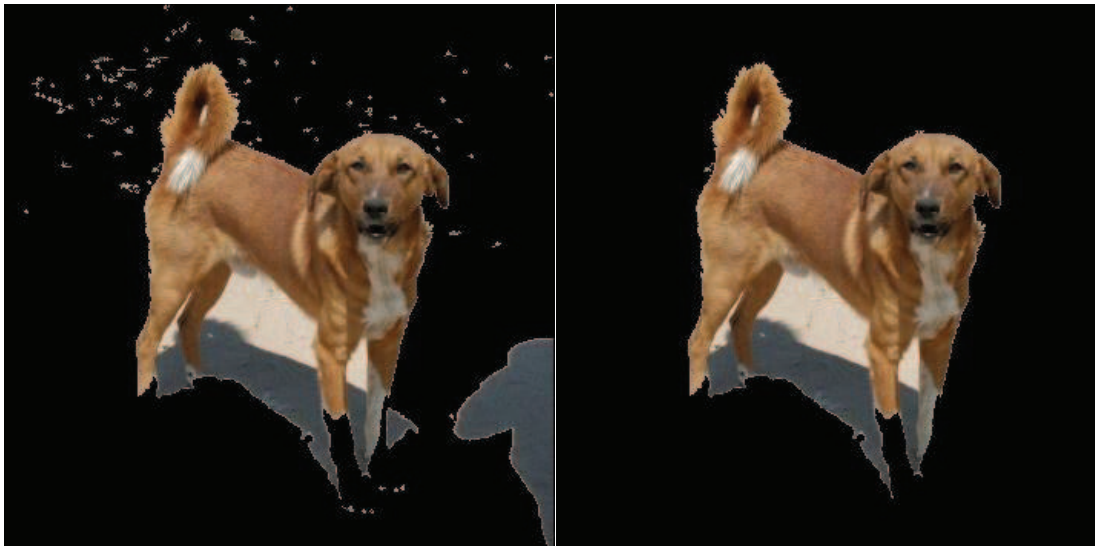
Obr. 6.9: Výsledek segmentace metodou semínko. Použit barevný model HSV(vlevo) a Lab(vpravo) při volbě semínka v okolí s podobností všech složek  $\pm 5$ .



Obr. 6.10: Výsledek segmentace metodou semínko. Použit model RGB a podobnost obrazových složek s velikostí  $\pm 2$ (vlevo) a  $\pm 10$ (vpravo).

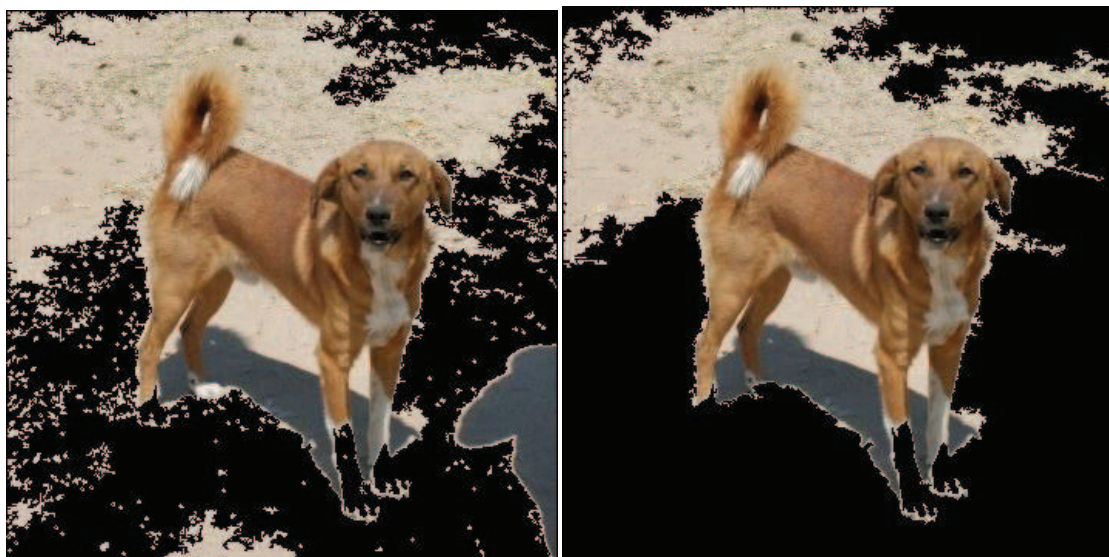
### 6.3 Segmentace založená na metodě kontur

Tato segmentační metoda pracuje s šedotónovým snímkem, takže odpadá testování segmentační funkce na různé barevné modely, ale otestujeme si závislost na zvolení různé hodnotě prahu a taky obsahu, z kterého se kontury počítají.



Obr. 6.11: Výsledek segmentace metodou kontur. Nastaven práh 98 a obsah 0(vlevo), 300(vpravo).





Obr. 6.12: Výsledek segmentace metodou kontur. Nastaven práh 61 a obsah 0(vlevo), 300(vpravo).

## 6.4 Anaglyf

V této podkapitole si ukážeme výsledek a zobrazení anaglyfického 3D zobrazení. Po nasazení speciálních (anaglyfických) brýlí se nám snímek jeví jako prostorový.



Obr. 6.13: Výsledek zobrazení anaglyfu pro posunutí snímku o 20.

# ZÁVĚR

Cílem diplomové práce bylo vytvořit funkční aplikaci pro segmentaci obrazu a anaglyfického 3D zobrazení. V první fázi diplomové práce byl vytvořen přehled základních a pokročilých metod segmentace obrazu. S tím souvisí i kapitola předzpracování obrazu, která je neméně důležitá při výsledné segmentaci. Následovaná kapitolou barevné modely, které taktéž ovlivňují výsledek segmentace, neboť je v každém modelu obraz definován různými parametry. Jako např. při RGB jsou obrazové body reprezentovány poměrem barev červená, zelená, modrá a u modelu HSV jsou body reprezentovány pomocí barevného tónu, sytosti a jasové hodnoty. V další fázi byl vytvořen program SegAnag, který byl optimalizován na možnosti uživatele ovlivnit výsledek segmentace a taky anaglyfického 3D zobrazení.

U tvorby algoritmu segmentace se kladl důraz na variabilitu metod. Tedy na možnost ovlivnění výsledku segmentace uživatelem. Program SegAnag nakonec nabízí tři různé metody segmentace. Těmito metodami jsou segmentační metoda Kmeans, metoda semínka a metoda kontur.

U jednotlivých metod má uživatel možnost změnit vstupní parametry a přímo tak ovlivnit výsledek segmentace. U metody kmeans má možnost změnit barevný model vstupního obrázku, kde má možnost změnit barevné schéma do následujících modelů: RGB, HSV, YCrCb, XYZ, Lab, Luv, HLS. Dále může uživatel změnit váhy složek barevného modelu. Při tomto kroku mu dopomáhá možnost si zobrazit histogramy jednotlivých složek, které jsou barevně rozděleny a ukázány v samostatných oknech. Pro uživatele byla implementována funkce „Doporučení“, která uživateli sdělí nejefektivnější barevný model pro segmentaci a procentuální poměr. Tento poměr odpovídá nastavením vah, kdy každých 10% znamená +1 u jednotlivé obrazové složky. Provedené experimenty prokázaly, že doporučené váhy je vhodné použít jako výchozí a dále s nimi pracovat dle potřeby. K další úpravě může uživatel docílit změnou počtu shluků v obraze, nebo velikosti průměrovacího okna. Zvětšování okna použitého pro průměrování automaticky zvyšuje výpočetní náročnost.

U metody semínka má uživatel zase možnost použít různé barevné modely. Parametrem segmentace je velikost prahů podobnosti ve vertikálním i horizontálním směru. Dále je implementována funkce zadávání bodu, buď v segmentovaném objektu, nebo v jeho okolí, kdy se posléze odstraní toto okolí a je zobrazen objekt jako hledaný segment.

U metody kontur uživatel pracuje s šedotónovým snímkem, kdy má možnost měnit práh pixelů, s kterých se počítají kontury, a velikost obsahu který má mít zobrazovaná kontura.

U anaglyfu uživatel stanoví jenom volbu posunutí snímku o určenou vzdálenost, a tím rozhodne jak mají objekty vystupovat ze snímku.

V kapitole 6 jsou ukázány výsledky programu SegAnag pro různé metody a nastavení jejich parametrů. U metody kmeans je hlavní parametr nastavení barevného modelu a následně jeho vah. Pro špatně zvolený obrazový model a jeho váhy má za následek, že výsledek segmentace se neshoduje s reálným objektem jako např. v obr. 6.2 u modelu HLS s konstantní vahou složek, naopak pro obr. 6.3 je výsledek modelu HLS diametrálně odlišný a můžeme rozlišit v našem případě objekt (pes). Důležitým

parametrem je také volba počtu shluku, který nám roztřídí obrazové body do tříd. Na obr. 6.4 je vidět důsledek použití pro různě shluky, kdy tři nám objekt vysegmentují, ale přidají do této skupiny i okolí nepatřící tomuto objektu. Naopak při velkém počtu nám rozdělí objekt na víc segmentů. Posledním parametrem, který ovlivňuje výsledek segmentace je volba průměrovacího okna. To má za následek při malé šířce odstranění některých částí z hledaného segmentu (zadní část psa a hlava). A naopak velké průměrovací okno nám objekt značně zmenší a chybějící části objektu jsou větší, některé zaniknou úplně (nohy psa).

U metody semínko závisí hlavně na zvoleném semínku, ze kterého se nám začíná metoda vypočítávat. To se určuje levým tlačítkem myši. Důležité je také v jakém barevném modelu se metoda aplikuje. Různé výsledky jsou vidět na obr. 6.9, na kterém jsou použity stejné meze podobnosti všech obrazových složek a zvoleno klikání na okolí kdy při zadání stejných výchozích bodů pozorujeme odlišné změny (mezi končetinami psa a nad psem). Vliv nastavení meze podobnosti obrazových složek je ukázáno na obr. 6.10, kdy při malých hodnotách se při zvolení bodu v okolí vysegmentuje příliš mnoho částí obrazu a při velké podobnosti sousedních pixelů se ztrácí skoro celý objekt. Při použití bodu do objektu nastává opačný problém. I tady si může trochu uživatel pomoci funkcí doporučit, kdy mu aplikace doporučí vhodný model a procentuelní poměr obsažený v obraze, tzn. když bude některá složka vyjádřena vyšším procentuelním vyjádřením tak přisoudíme této obrazové složce větší míru podobnosti sousedních pixelů.

U segmentace založené na metodě kontur závisí hlavně na zvoleném prahu, který vyjadřuje hodnoty jasu, z kterých se budou hledat kontury. Na obrázcích 6.11 a 6.12 vidíme vliv tohoto prahu, když ho nastavíme dostatečně velký vysegmentuje nám objekt v obraze, naopak při nedostatečné velikosti prahu nám vysegmentuje i části nesouvisejícím s hledaným objektem. Volbu výsledku segmentace můžeme ovlivnit i nastavením velikosti obsahu kontur, které se budou zobrazovat. Při nulové hodnotě se zobrazí všechny nalezené kontury v obraze, ale s postupným zvyšováním kontury s malým obsahem mizí a zůstává v ideálním případě náš hledaný objekt.

V poslední části aplikace může uživatel zobrazit anaglyfický 3D snímek, který zobrazí snímek jako na obr. 6.13. Když využijeme speciální (anaglyfické) brýle, tak se nám objekt ve snímku jeví jako prostorový. Pro velikost prostorového efektu může uživatel volit různé hodnoty. Dle provedených testů je vhodné volit parametr posunutí od hodnot 5 do hodnoty 30.

Aplikace byla vytvořena programovacím jazykem C++ s knihovnou openCV v prostředí Visual Studio 2008. To má za následek, že program je spustitelný na počítačích s operačním systémem Windows. Nicméně k jeho správné funkci musí mít uživatel nainstalován balíček Microsoft .NET Framework, který ale bývá standardně součástí operačního systému.

# LITERATURA

- [1] JÁN, J. Medical Image Processing, Reconstruction and Restoration - Concepts and Methods. Boca Raton, FL, USA: CRC Press, Taylor and Francis Group, 2006. 730 s. ISBN 0-8247-5849-8.
- [2] KRUMPHOLC, L. *Metody segmentace biomedicinských obrazových signálů*. Diplomová práce. Brno: FEKT VUT v Brně, 2009.
- [3] Kalová, I. Počítačové vidění. Podpora k přednáškám předmětu MASO [online]. 2010 [cit. 2011-04-15]. Dostupný z WWW: <<http://www.dbme.feec.vutbr.cz/ubmi/courses/MASO/MASO09.pdf>>.
- [4] ŠPANĚL, M, BERAN, V. Obrazové segmentační techniky. Přehled existujících metod [online]. 2006 [cit. 2011-04-15]. Dostupný z WWW: <<http://www.fit.vutbr.cz/~spanel/segmentace/>>.
- [5] KALOVÁ, I. Úpravy obrazu. Detekce a parametrizace geometrických tvarů – Houghova transformace [online]. 2007 [cit. 2008-12-09]. Dostupný z WWW: <<http://www.uamt.feec.vutbr.cz/vision/TEACHING/MPOV/07%20-%20Detekce%20a%20parametrizace%20geometrickych%20tvaru.pdf>>.
- [6] JÁN, J. Analýza signálů a obrazů. Podpora k přednáškám předmětu MASO [online]. 2010 [cit. 2011-04-15]. Dostupný z WWW: <<http://www.dbme.feec.vutbr.cz/ubmi/courses/MASO/>>
- [7] MUDROVÁ, M. Barvy v počítačové grafice. Podpora k přednáškám předmětu Zpracování obrazů [online]. 2004 [cit. 2011-12-15]. Dostupný z WWW: <<http://uprt.vscht.cz/mudrova/zob/prednasky/02-BARVY/Barvy.pdf>>
- [8] JANČÍK, Z. Vyukový program pro demonstraci principu barev a barevných modelů. [online]. 2006 [cit. 2011-12-15]. Dostupný z WWW: <<http://janosik.zlutaponorka.com/www-bp/>>
- [9] PETYOVSKÝ, P. Barevné modely. Podpora k přednáškám předmětu MPOV [online]. 2010 [cit. 2011-12-15]. Dostupný z WWW: <<http://www.uamt.feec.vutbr.cz/vision/TEACHING/MPOV/01c%20-%20Barevne%20modely.pdf>>
- [10] Wismüller A., Vietze F. et al.: *A Neural Network Approach to Adaptive Pattern Analysis - the Deformable Feature Map*. European Symposium on Artificial Neural Networks, ESANN'2000, pp. 189-194, Bruges, Belgium, April 2000.
- [11] Novák M. a kol.: *Umělé neuronové sítě, teorie aplikace*. Praha: C. H. Beck, 1998.
- [12] MAŠKA, M. Segmentace obrazu pomocí rychlých aproximací implicitních aktivních povrchů. [s.l.], 2006 [cit. 2008-12-09]. 53 s. Masarykova univerzita, Fakulta informatiky. Diplomová práce. Dostupný z WWW: <[http://is.muni.cz/th/60734/fi\\_m/print.pdf](http://is.muni.cz/th/60734/fi_m/print.pdf)>.
- [13] JANOUC, M. Zpracování obrazu jednočipovým mikroprocesorem. Diplomová práce. Praha: FEL ČVUT v Praze, 2008.
- [14] POYNTON, Ch. Frequently Asked Questions about Color [online]. 2010 [cit. 15.12.2011]. Dostupný z WWW: <<http://www.poynton.com/PDFs/ColorFAQ.pdf>>
- [15] Bradski, G.; Kaehler, A.: *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, první vydání, September 2008, ISBN 0596516134.

- [16] FRANC, V. Shlukování k-means [online]. 2001 [cit. 15.12.2011]. Dostupný z WWW:  
<[http://cmp.felk.cvut.cz/cmp/courses/recognition/Lab\\_archive/RPZ\\_02-03l/kmeans/index.html](http://cmp.felk.cvut.cz/cmp/courses/recognition/Lab_archive/RPZ_02-03l/kmeans/index.html)>

## SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

$u, v$	Jasová stupnice obrazu
$g(i, j)$	Výstupní obraz
$f(i, j)$	Vstupní obraz
$D$	Množina prahů
$G(i, j)$	Výsledný gradientní obraz
$G_R(i, j)$	Gradient po detekci hran pro horizontální směr
$G_S(i, j)$	Gradient po detekci hran pro vertikální směr
$\Delta^2 f(j, k)$	Laplasián
$\sigma^2()$	Rozptyl, druhá derivace
$H(x, y)$	výstupní hranový obraz
$G(x, \sigma)$	Gausovská funkce
$G(y, \sigma)$	Gausovská funkce
$\rho$	normalizovaná vzdálenost přímky od počátku
$\theta$	úhel vzhledem k ose x
$a, b, p$	parametry kružnice
$T$	kritérium
$p_s$	vstupní pixel
$p_j p_i$	pixely
$p_a$	střední hodnota pixelů
$X(), Y()$	parametry funkce
$s$	skalární proměnný parametr
$p_0$	vektor ovlivňující tvar křivky
$\Phi()$	vrstevnice proměnného skalárního pole
$N$	Normálový vektor
$\nabla$	gradient
$V_0$	tlaková tendence
$E_N$	energie vnitřní kontury
$E_I$	energie obrazu
$E_T$	energie omlazení